



# Motivational Agile Metrics

## 改善をモチベートするメトリクスでチームが変わる

IPAセミナー (3/12(火))  
～組込みソフトウェア開発 定量データ活用推進セミナー～

# 自己紹介

永田 敦

サイボウズ株式会社

開発本部

アジャイルコーチ

WingArc1ST



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

Agile流派 Evolutionary (EVO)





ビジネスアプリ作成クラウド

かんたんに  
作れる

プログラミングも特別  
なスキルも不要

高度な  
カスタマイズ

システム連携や、高度  
な開発にも対応



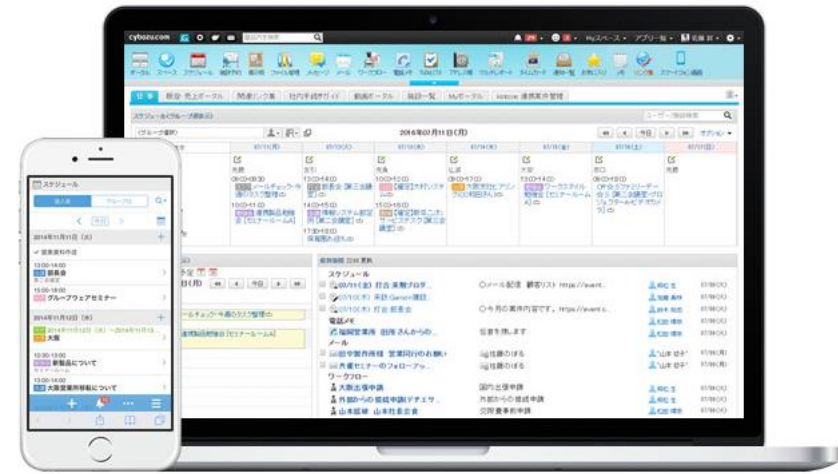
大企業向け管理機能を搭載

ユーザー数  
300から

300名から数万人規模  
の組織に最適

大企業に  
最適化

詳細な権限設定、国際  
化、システム連携対応



開発の管理、要求管理、業務オペレーションを自社製品で行っている

# アジェンダ

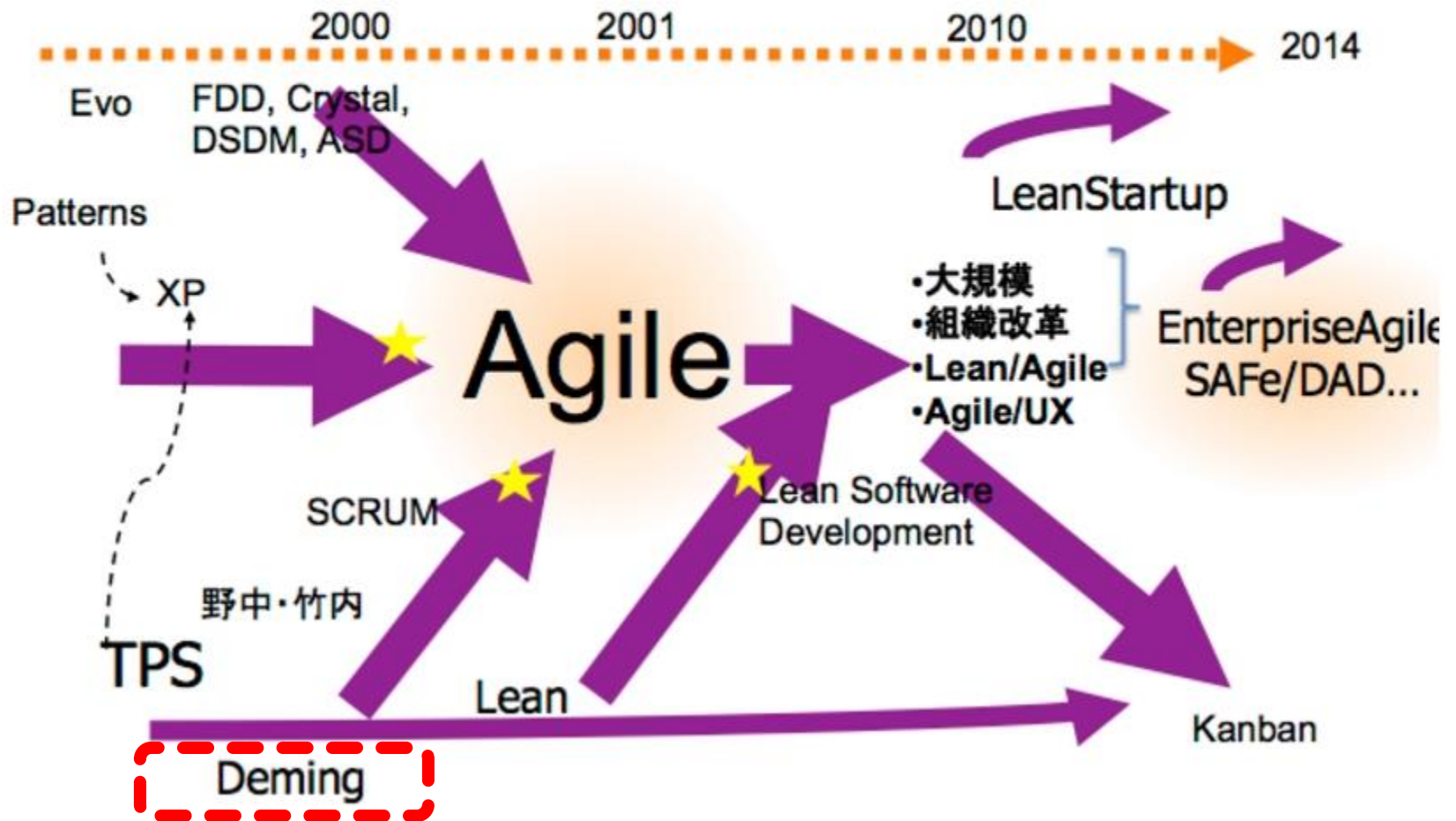
---

- ▶ **アジャイル開発へのモチベーション**
- ▶ **PDSAサイクル**
- ▶ **スクラムフレームワーク**
- ▶ **OODA**
- ▶ **Motivational Agile Metrics**
- ▶ **まとめ**

# アジャイル開発へのモチベーション

プロジェクトを成功させたい

# アジャイルの現在位置



An Agile way, kenji Hiranabe, <https://anagileway.wordpress.com/2015/12/04/what-is-enterprise-agile-1/>

# プロジェクトの成功率

MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

*The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011-2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.*

Standish Group 2015 Chaos Report



# プロジェクトの成功率 アジャイル対ウォーターフォール

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL				
SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011–2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.



# プロジェクトは失敗する

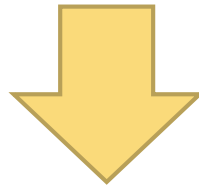
---

- ▶ **ソフトウェア開発プロジェクトは毎回違う**
  - ▶ 要求と価値
  - ▶ 必要とされる技術
  - ▶ 環境（内部環境、外部環境）
  - ▶ 組織（チーム）
- ▶ **ソフトウェアは目に見えない**
- ▶ **初めから本当の要求を正確に獲得し、表現伝達することが非常に難しい**
- ▶ **それに必要な情報が初めからわかっているわけではない**
- ▶ **初めから、すべてのリスクを認識することが難しい**

# プロジェクトの失敗から学ぶ

---

ベストプラクティスからは学べない  
銀の弾丸はない



自らのドメイン、コンテキストから、  
経験的に学んでいく必要がある

エンピリカル・プロセス

# エンピリカル・プロセス

---

## 今日のソフトウェア

サイズが著しく増えた

例:自動車 1000万行  
20年間で100倍

複雑度が著しく増えた



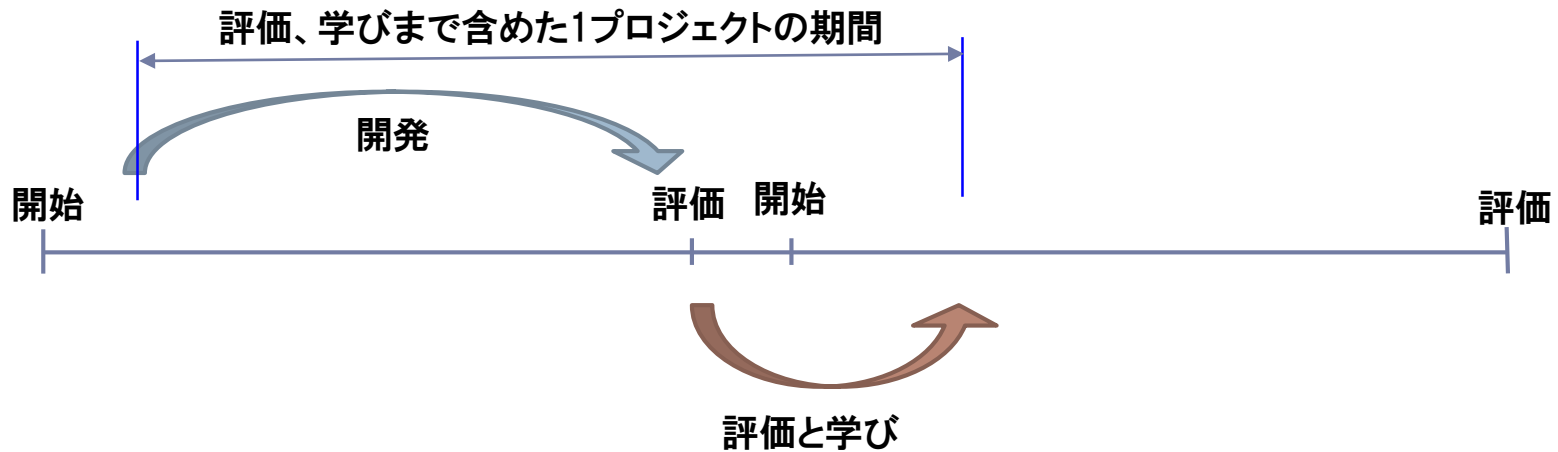
不確実性の増大

プロジェクトのすべてを予測し、計画することが困難

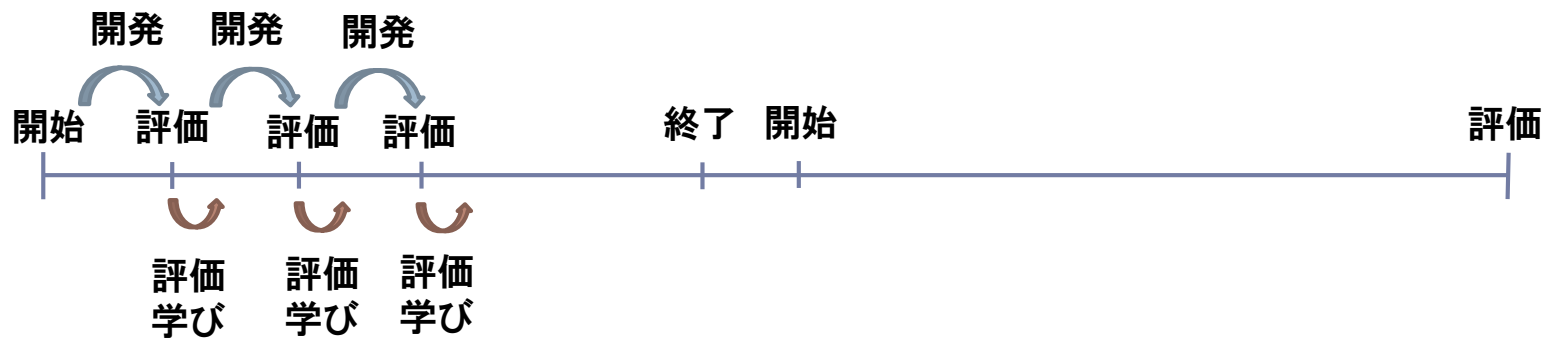
プロジェクトを逐次評価し、アダプティブに改善する

# プロジェクトの評価と学び

## ウォーターフォール



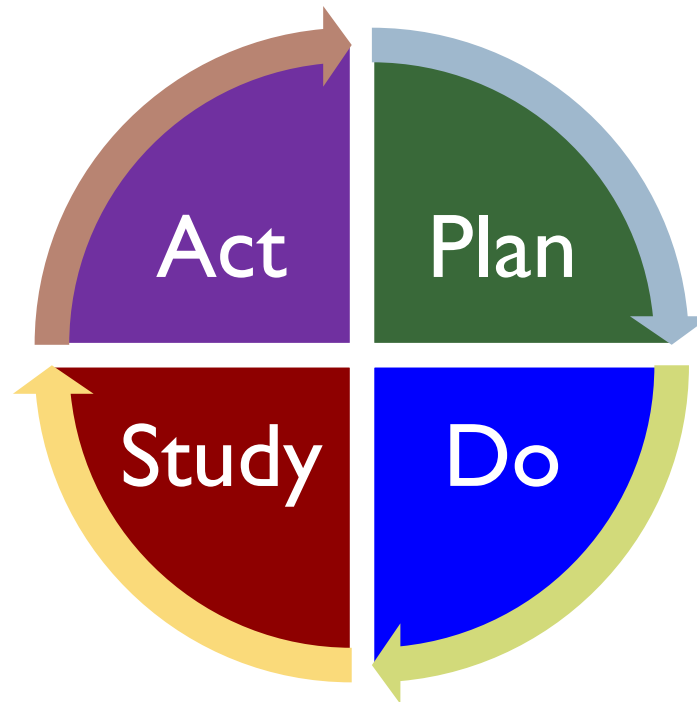
## アジャイル



How quality is assured by Evolutionary Method, Niels Malotau, 2014

# PDSA イテレーティブに学習するしくみ

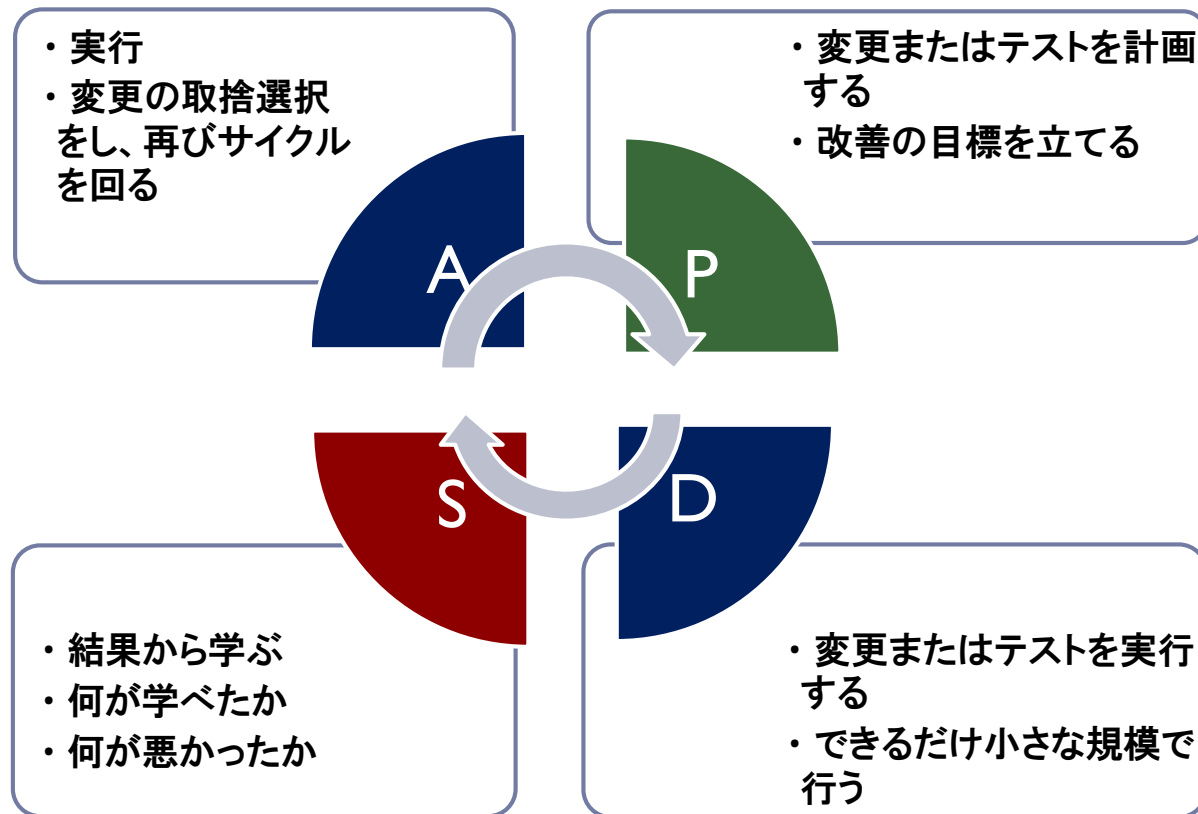
---



# PDSAサイクル

デミングの改善の心

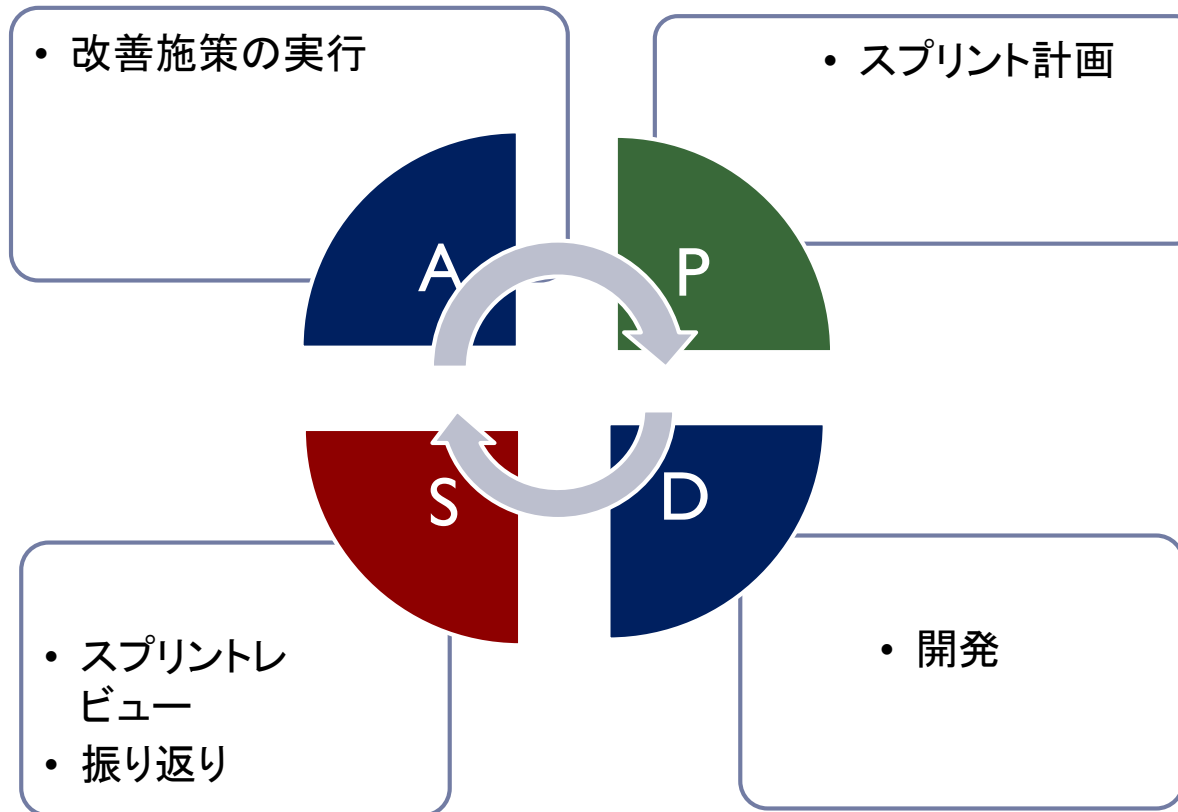
## PDSA(Plan-Do-Study-Act) Cycle 1993





# Deming PDCA物語：PDSAサイクルとアジャイル

## 例えばSCRUM

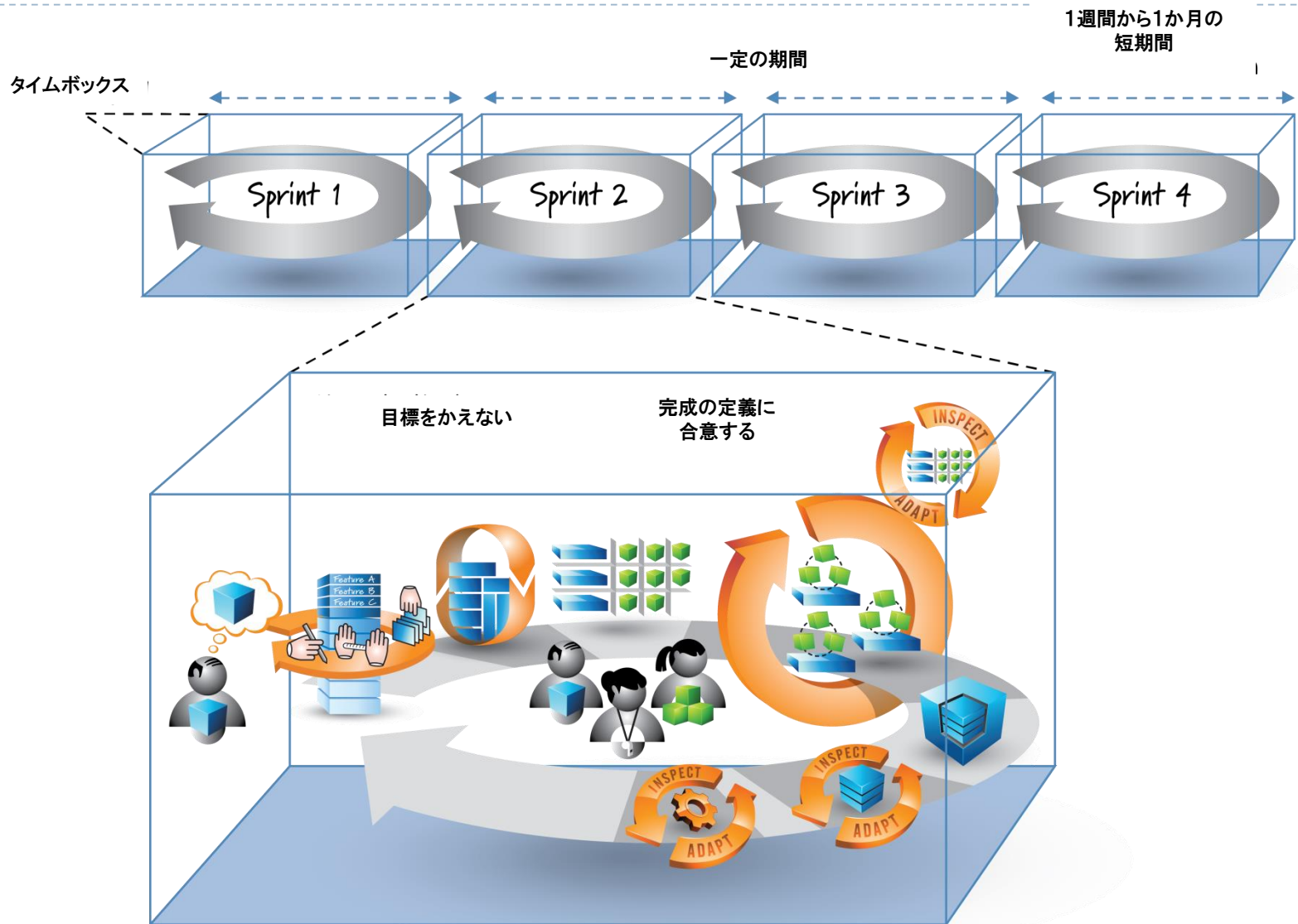


## デミングの考えはアジャイルのルーツ

# スクラムフレームワーク

アジャイル開発の代表的フレームワーク

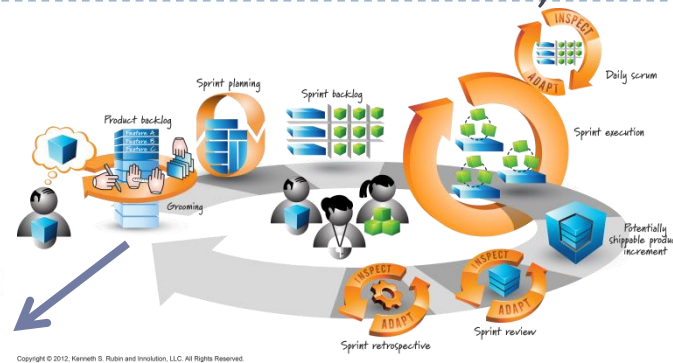
# スプリントの概要



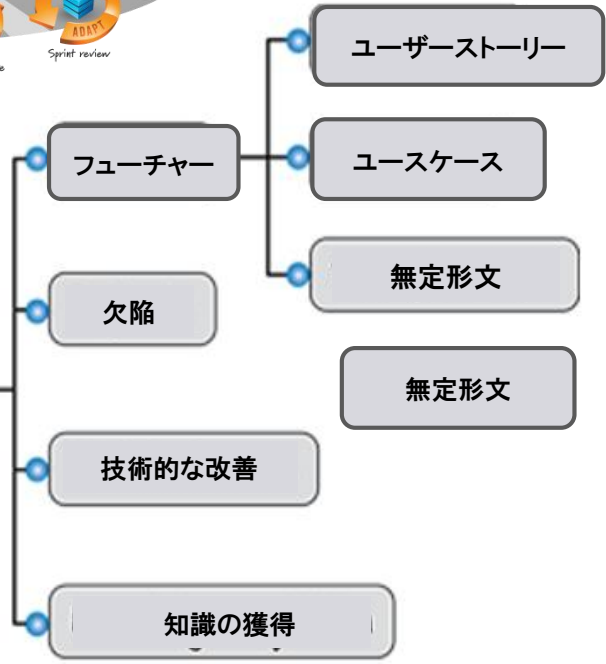
# プロダクトバックログと プロダクトバックログアイテム (PBI)

## プロダクトバックログ

アイテム | サイズ



プロダクトバックログアイテム  
(PBIs)

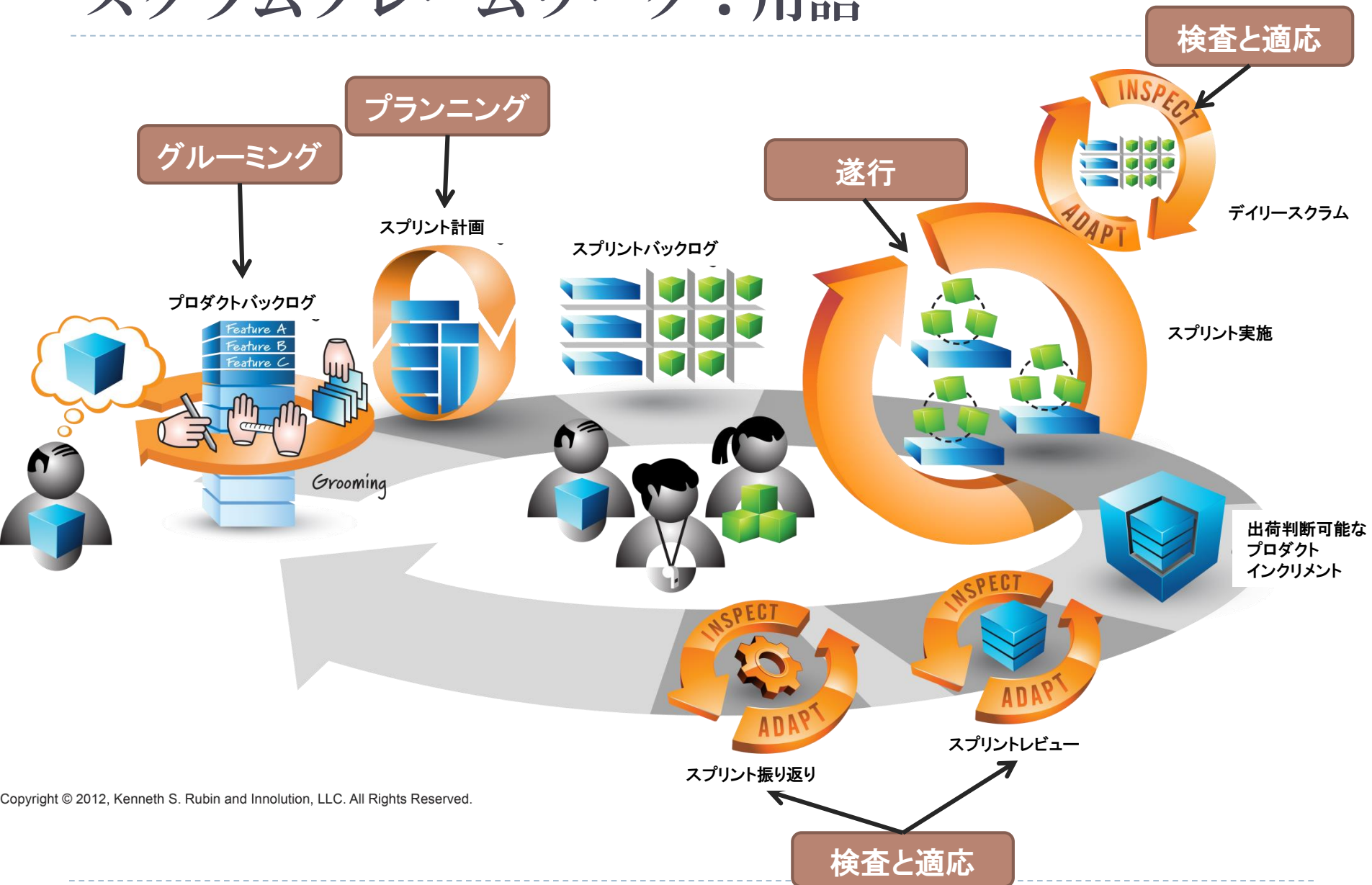


# 用語

---

- ▶ **プロダクトバックログ**
  - ▶ プロダクトに必要なものがすべて上げてあるリスト
- ▶ **プロダクトバックログアイテム (PBI)**
  - ▶ プロダクトバックログの項目
- ▶ **スプリントバックログ**
  - ▶ プロダクトバックログの項目を完成させるために必要な作業の計画
- ▶ **インクリメント**
  - ▶ 成果物としてのコード
  - ▶ イテレーションごとにインクリメンタルに成長していくのでインクリメントと呼んでいる
- ▶ **出荷判定可能なインクリメント**
  - ▶ 顧客に出してもよい品質レベルに仕上げたコード

# スクラムフレームワーク：用語



Copyright © 2012, Kenneth S. Rubin and Innolution, LLC. All Rights Reserved.

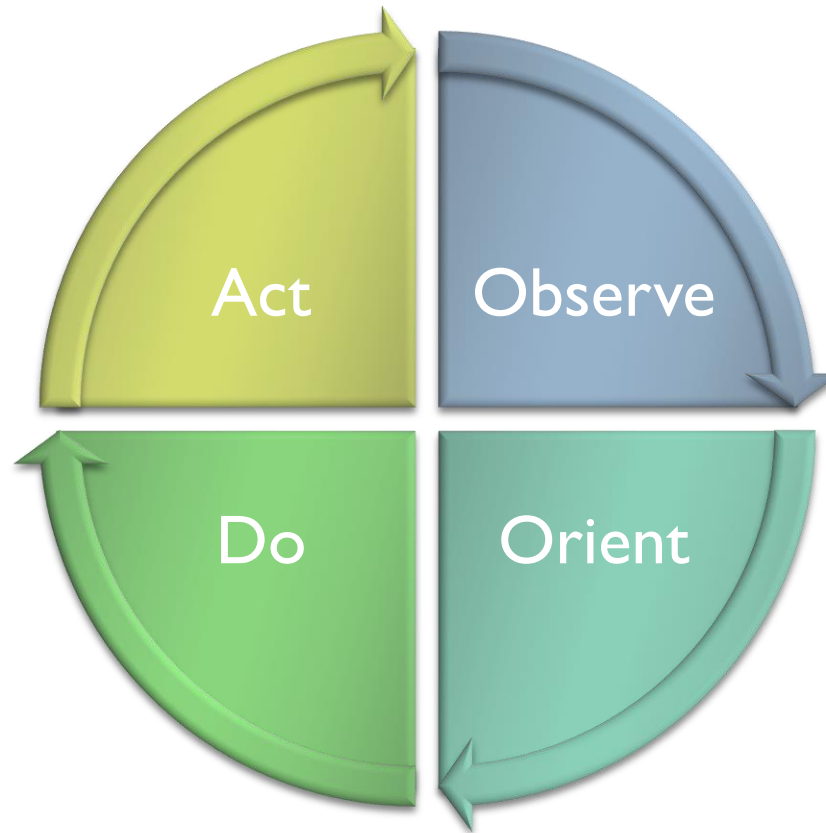
OODA

フィードバックの戦略

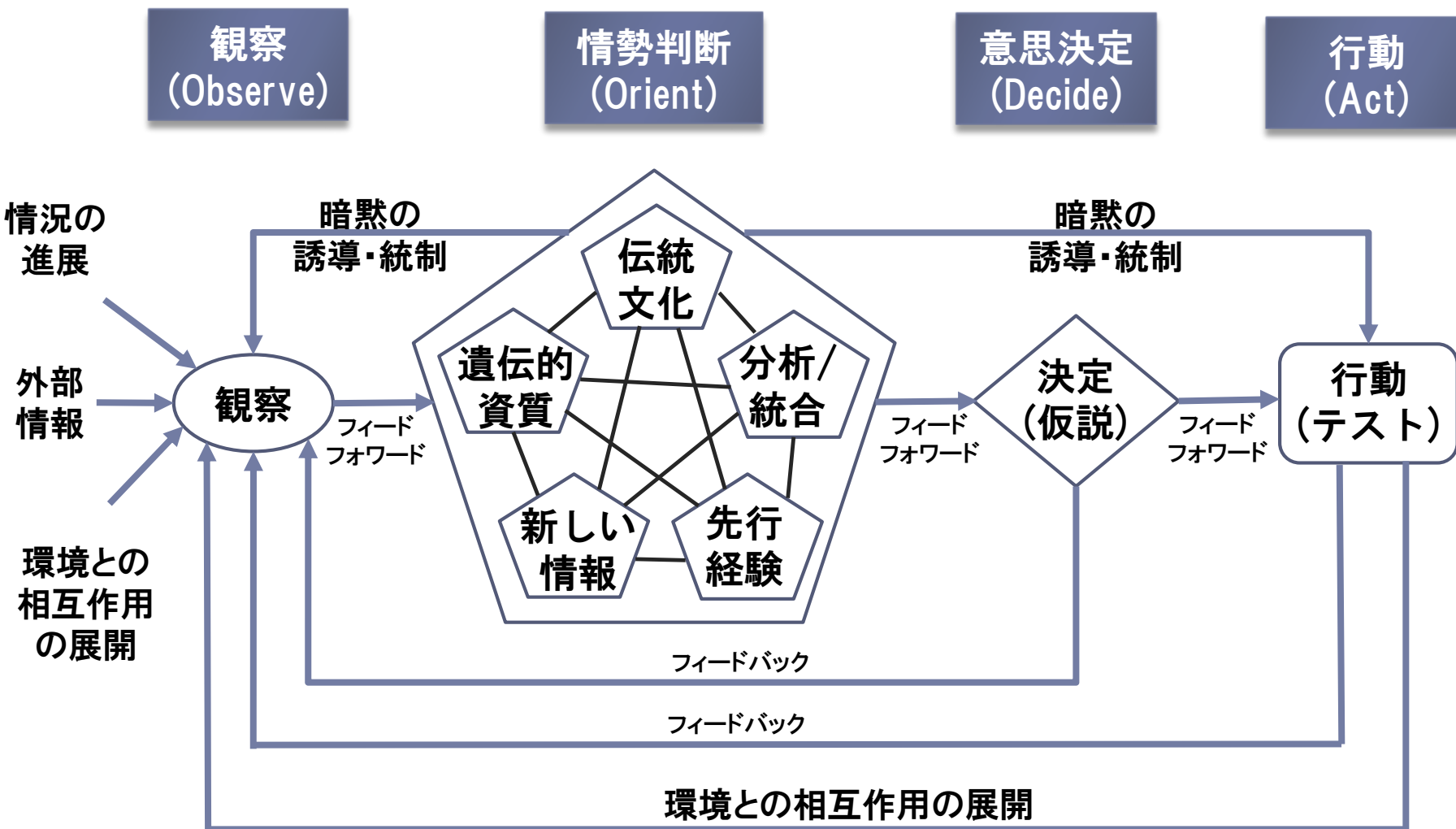


# OODA

---



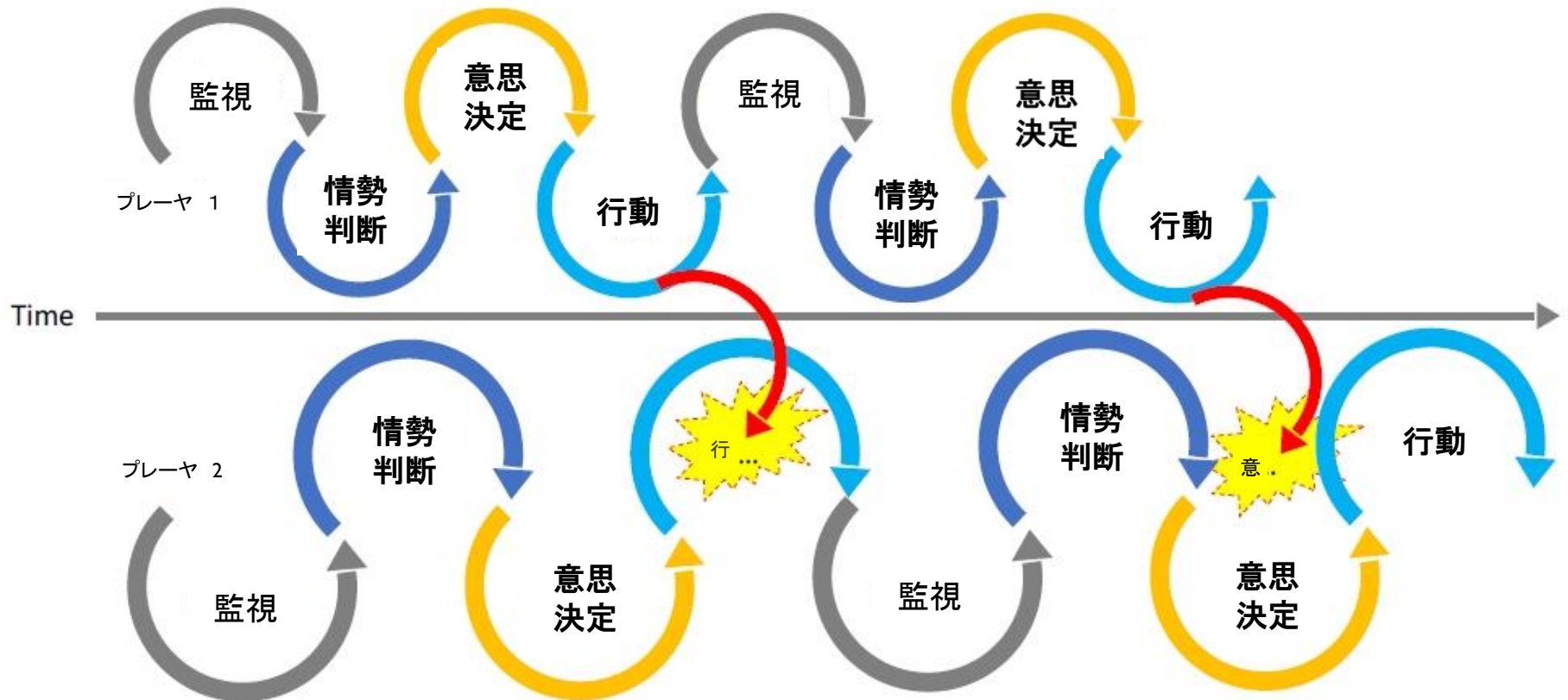
# OODAループ



野中, 郁次郎『知的機動力の本質 - アメリカ海兵隊の組織論的研究』中央公論新社、2017年。ISBN 978-4-120049-74-3。

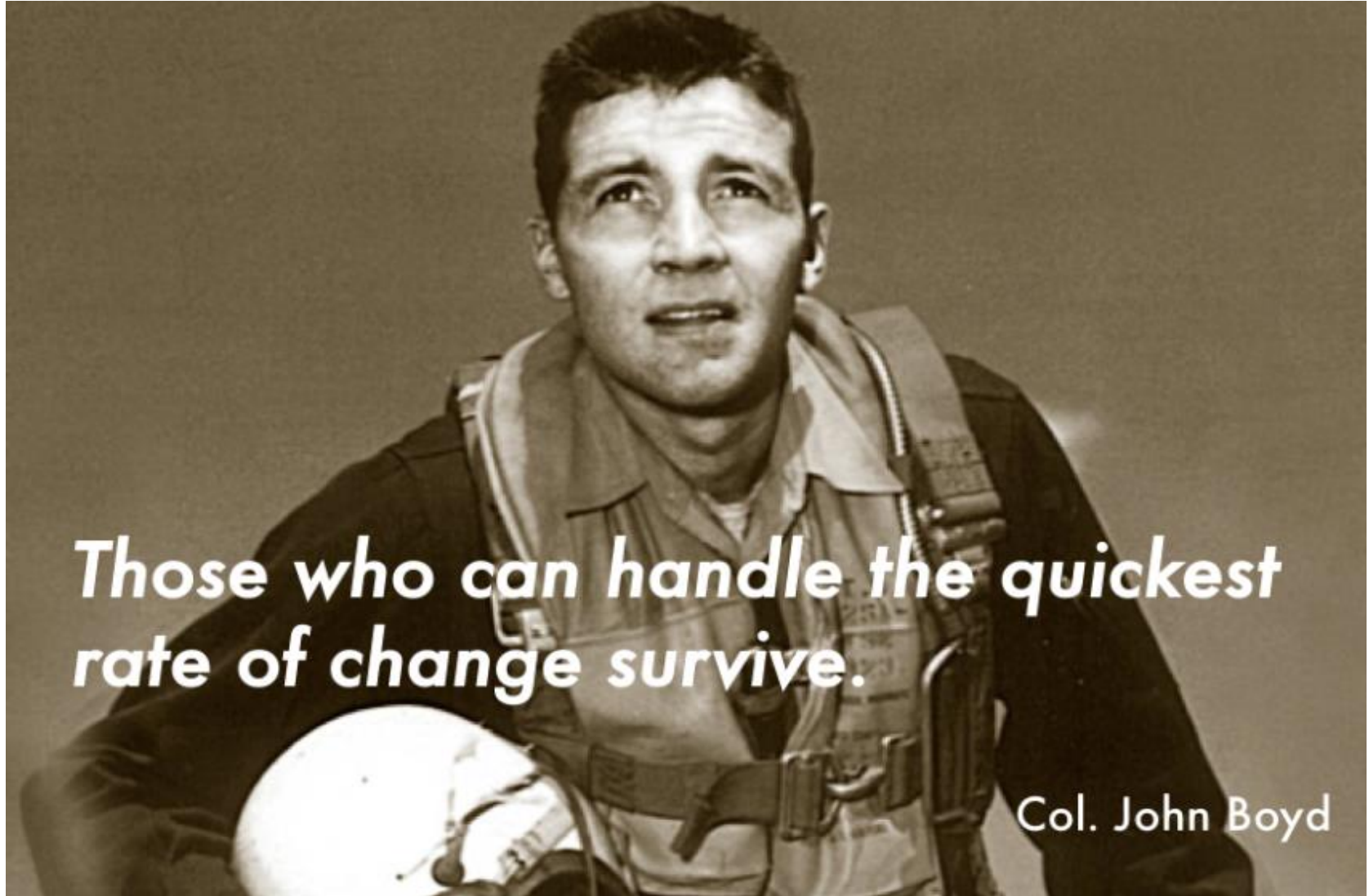
# 戦局を左右するのは情報量と意思決定のスピードである

## 相手のループの内側に入る…



# John Boyd

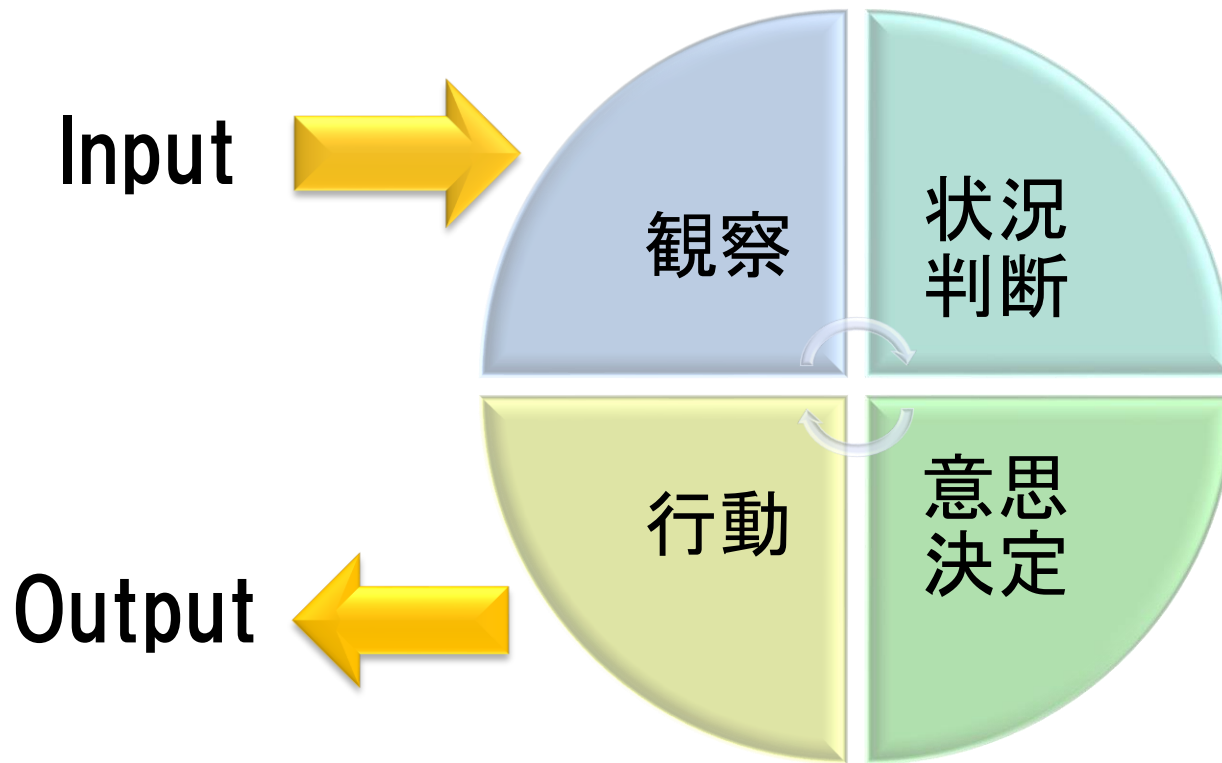
---



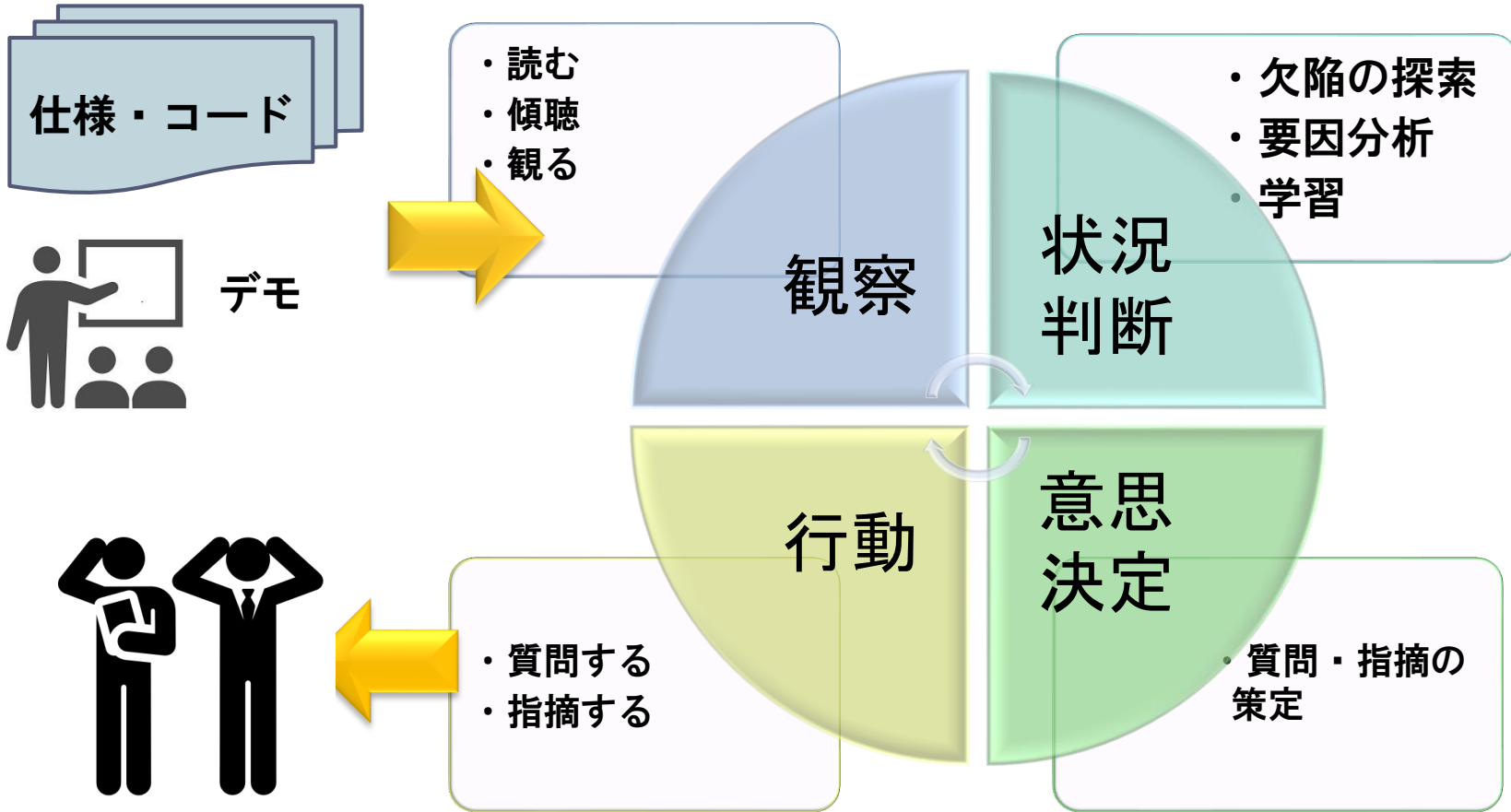
[/https://medium.com/@raldon/make-better-decisions-lessons-from-a-fighter-pilot-and-the-honey-badger-2258df453ed1](https://medium.com/@raldon/make-better-decisions-lessons-from-a-fighter-pilot-and-the-honey-badger-2258df453ed1)

# OODA ループ

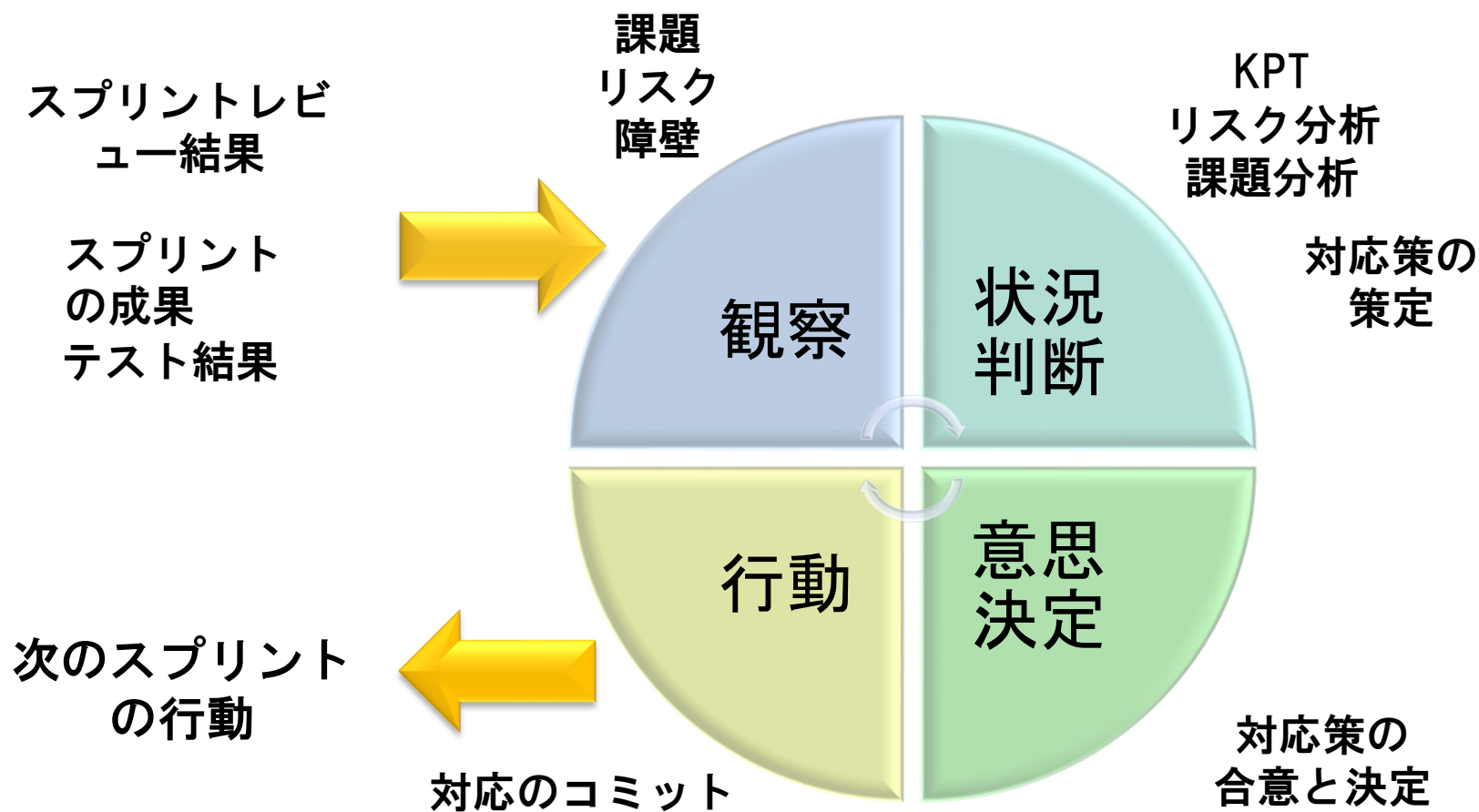
---



# レビュー

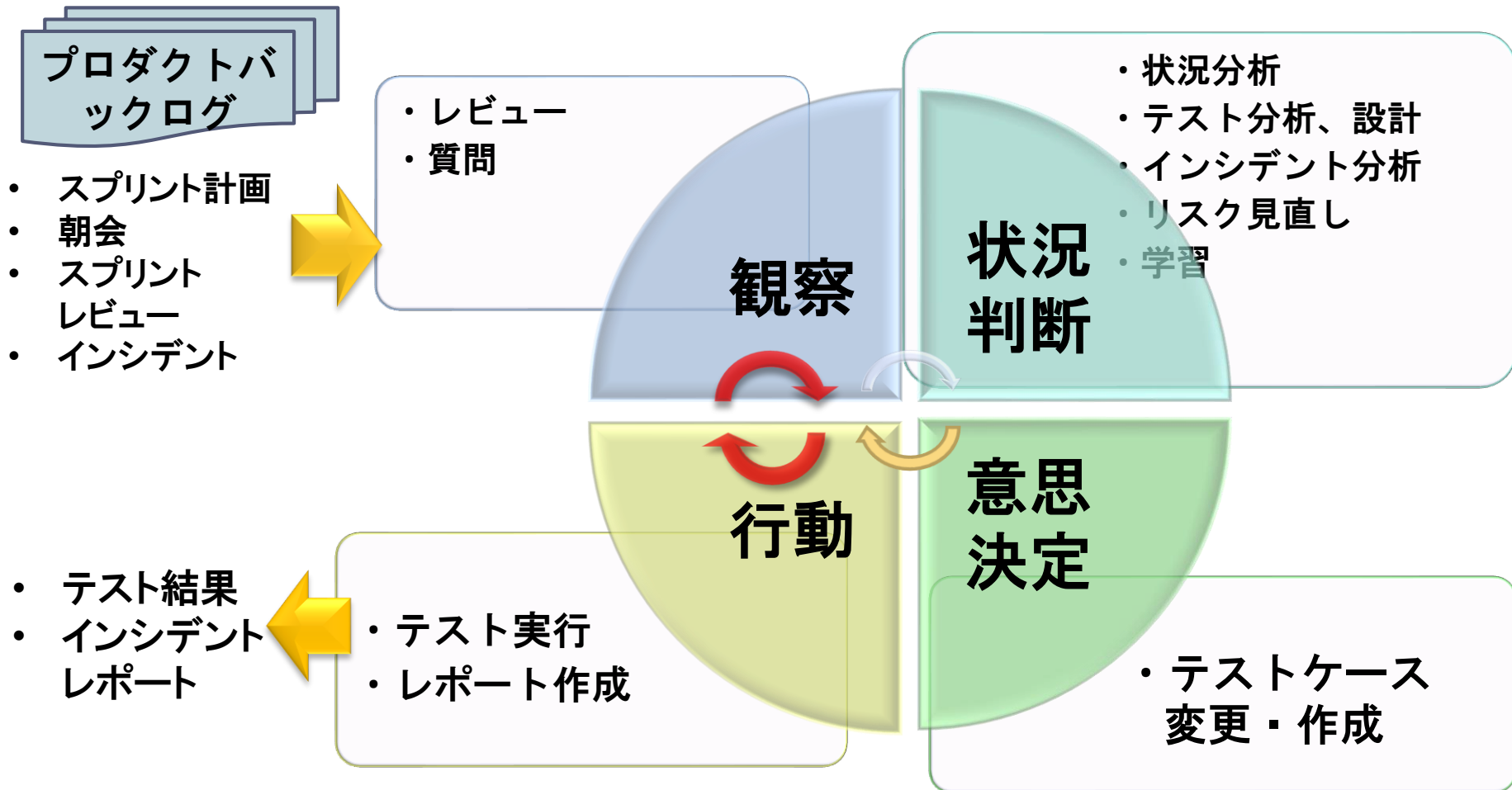


# 振り返り





# テスト



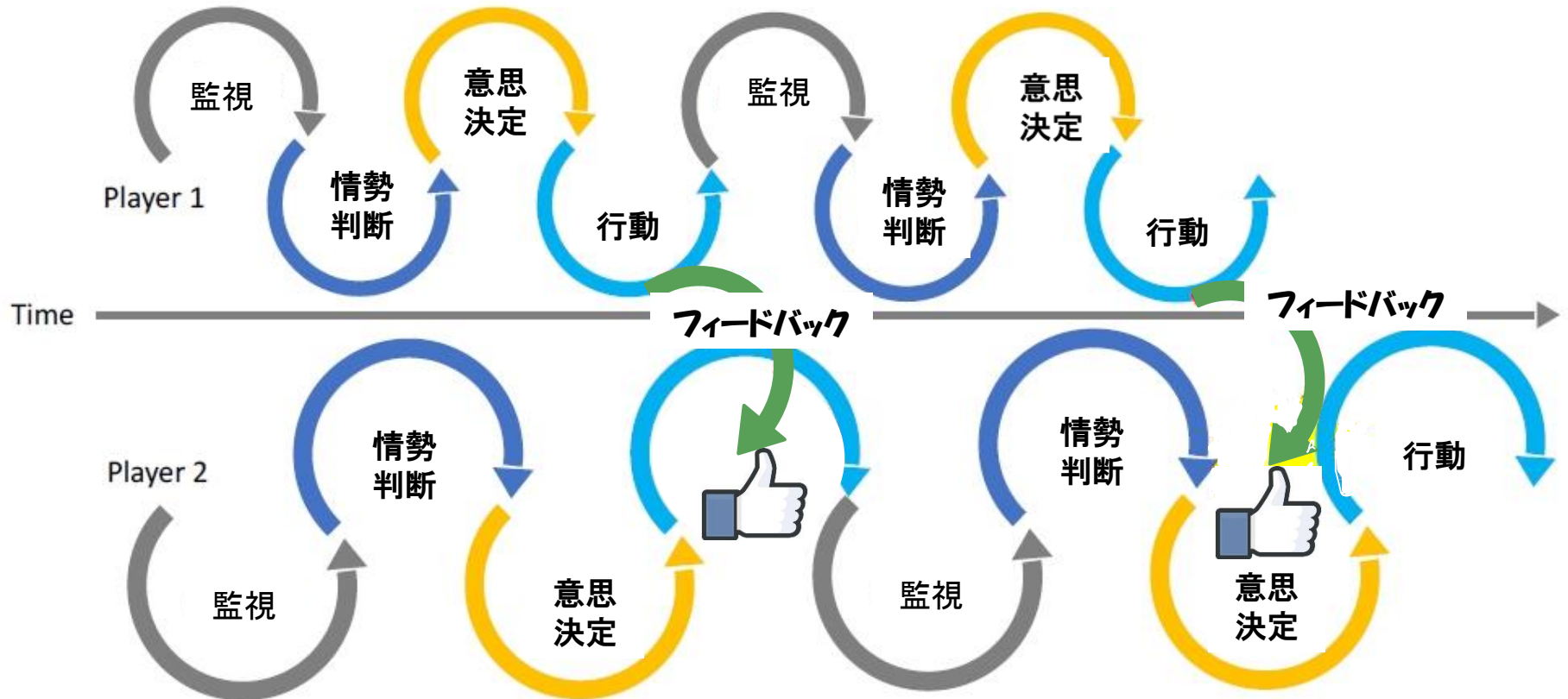
# 品質の見える化：メトリクス

---

- ▶ **メッセージを持ったメトリクス**
  - ▶ 受け取ったステークホルダが行動を起こすメッセージが必要
  - ▶ ゴール、目的
- ▶ **要素**
  - ▶ ストーリポイント
  - ▶ 工数時間
- ▶ **見せ方**
  - ▶ 二つの手法
    - ▶ メッセージを出すためにグラフを変えていく
    - ▶ 定点観測
      - 行動（改善）した効果を示す
  - ▶ バーンダウンチャート
  - ▶ バーンアップチャート
  - ▶ ベロシティ

# 適切なフィードバックによる改善

## 互いを自律的に改善し合う



# Motivational Agile Metrics

モチベーションのメッセージを持つメトリクスとその事例

# Motivational Metrics

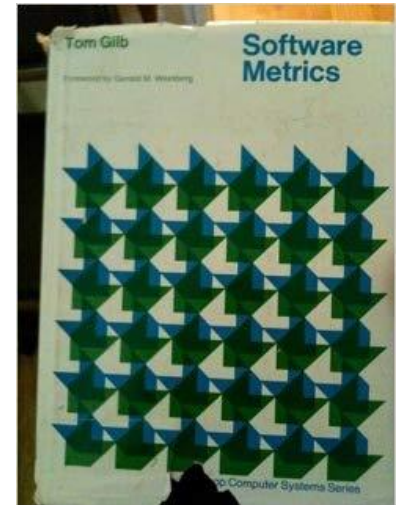
---

1976 Tom Gilb, “Software Metrics”

“”メトリクスはソフトウェア開発チームにシステムの品質を望まれるレベルにしていく意欲を起こさせる力がある”



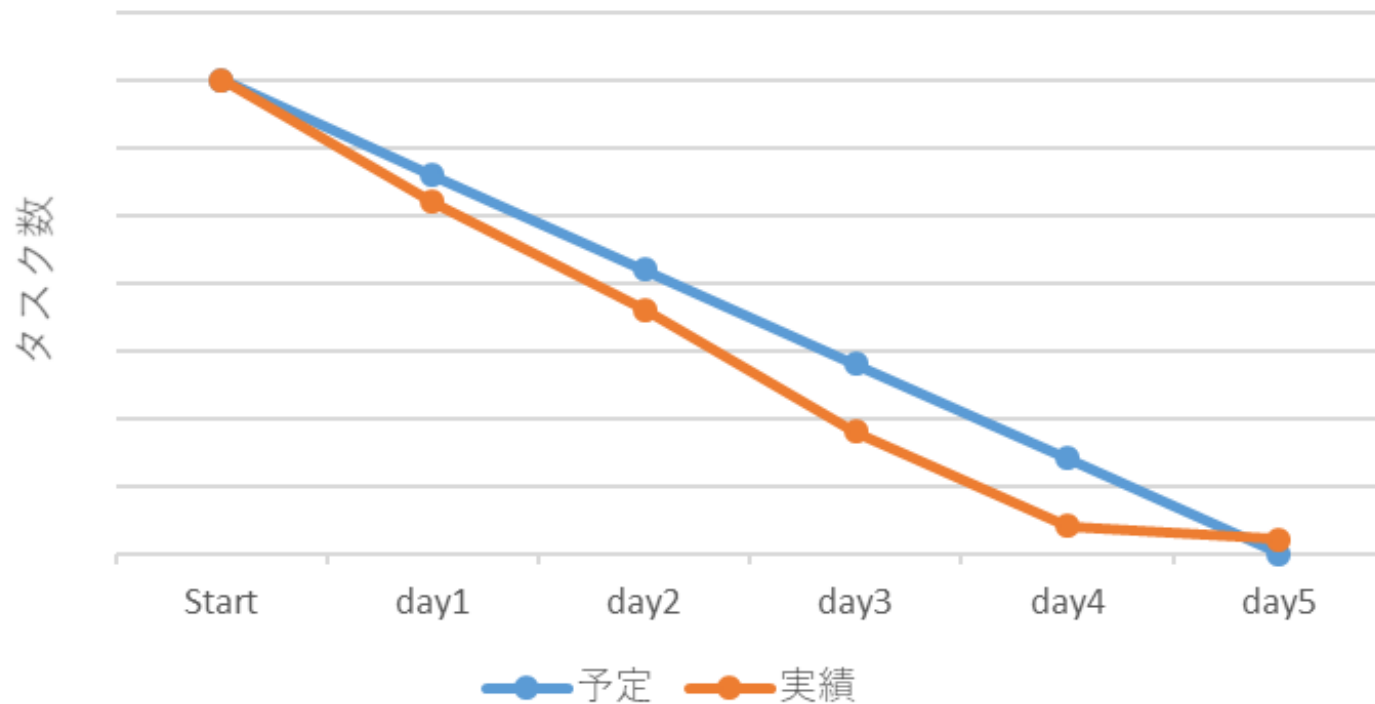
IIDについて明確に議論された初めての本  
この本によって、Tom Gilbはアジャイルの父ならぬアジャイルの祖父(Grand Father)と呼ばれる



Iterative and Incremental Development: A Brief History, Craig Larman, Victor R. Basili, 2003

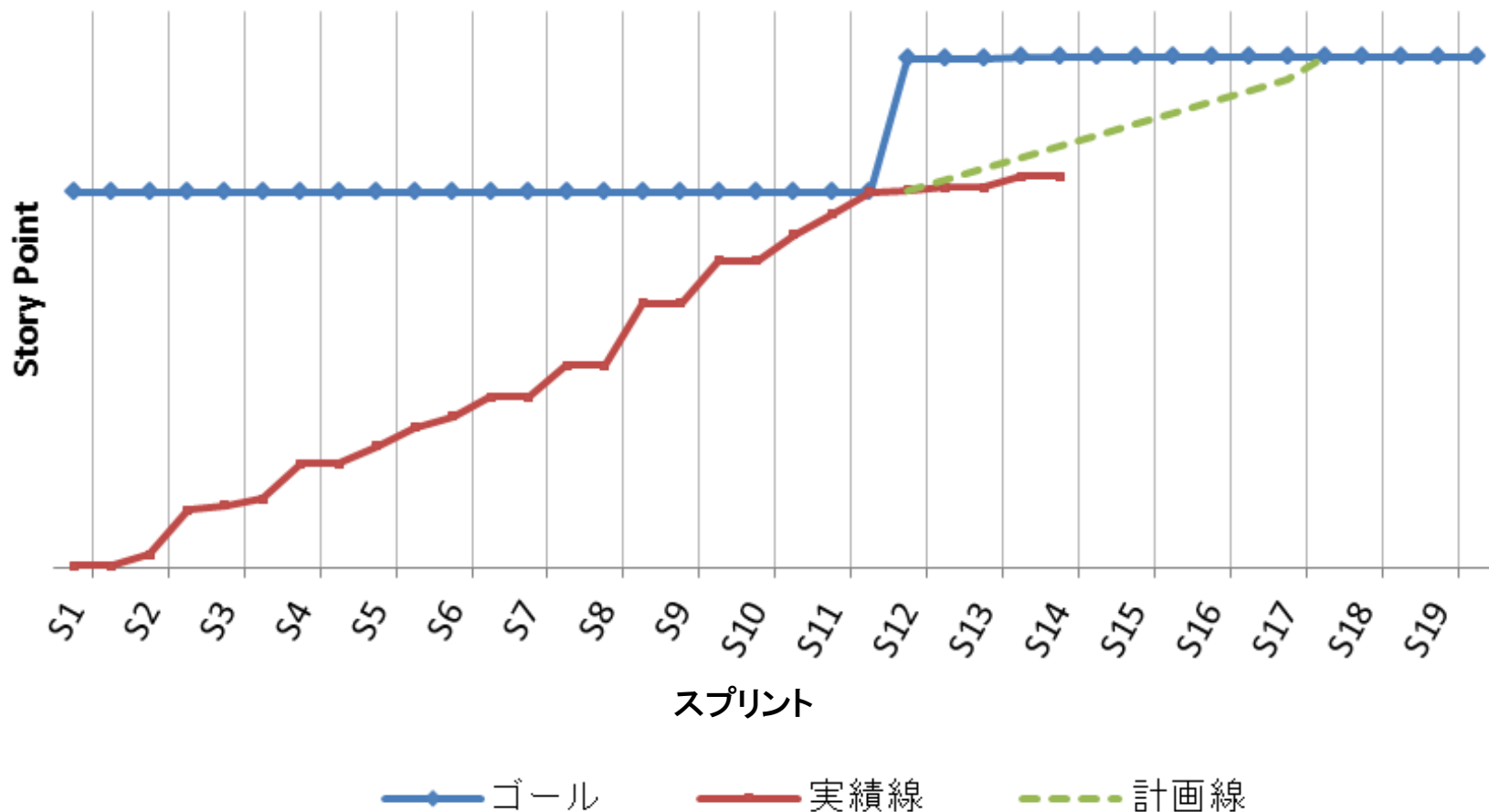
# バーンダウンチャート

バーンダウンチャート

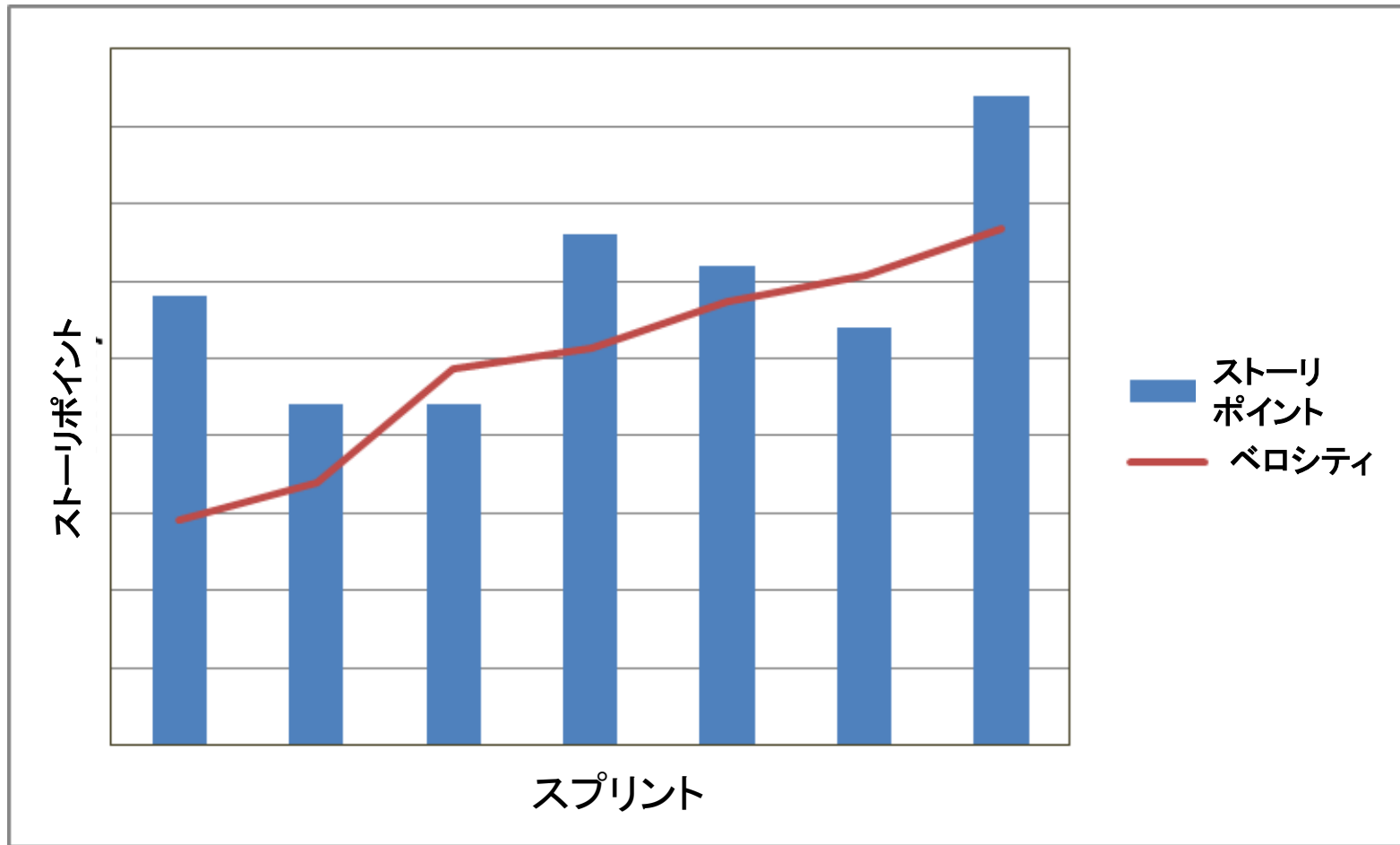


# メトリクス：バーンアップチャート

## バーンアップチャート

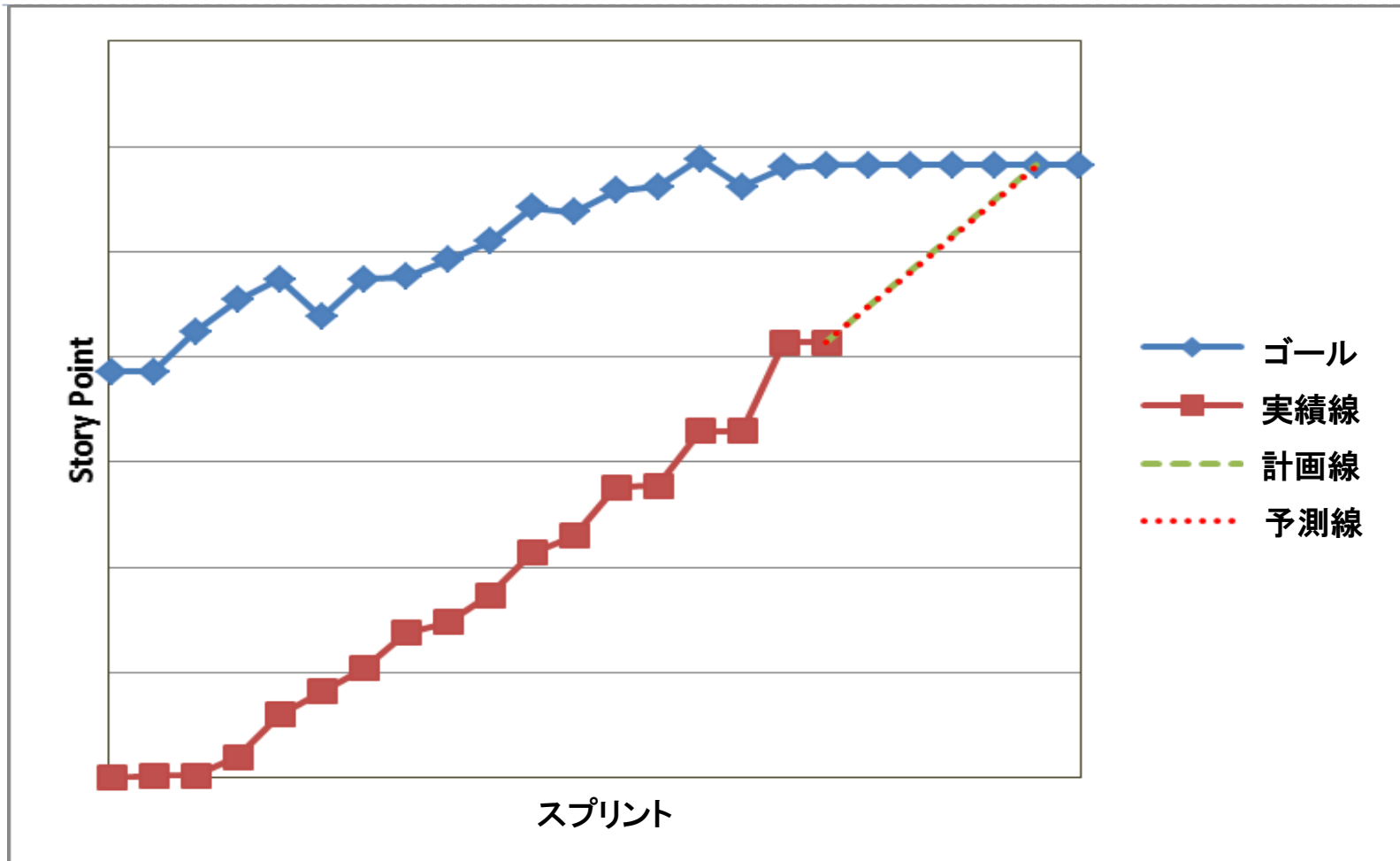


# ベロシティ: ストーリポイントの移動平均



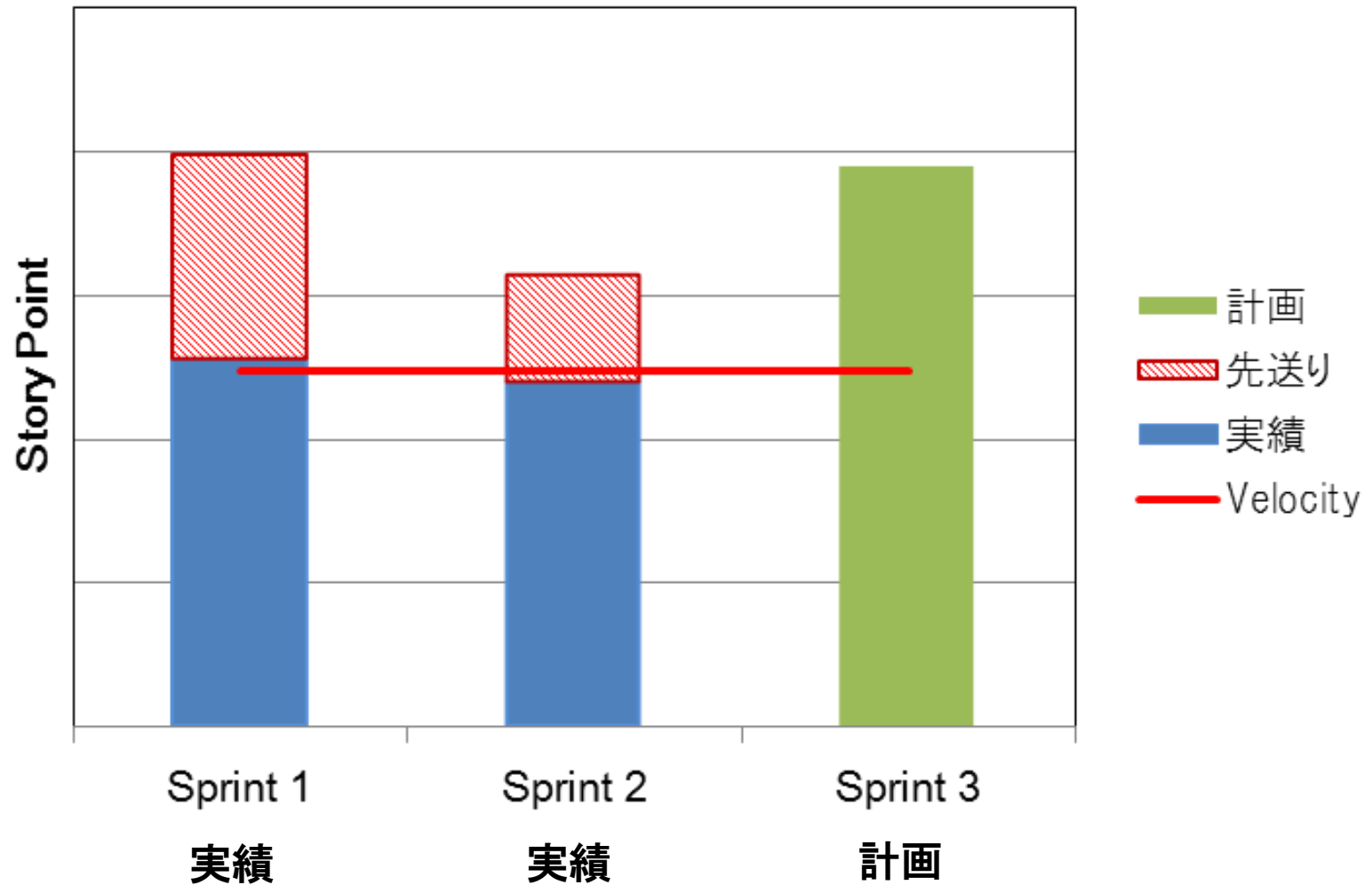


# メトリクス：実際のバーンアップチャート

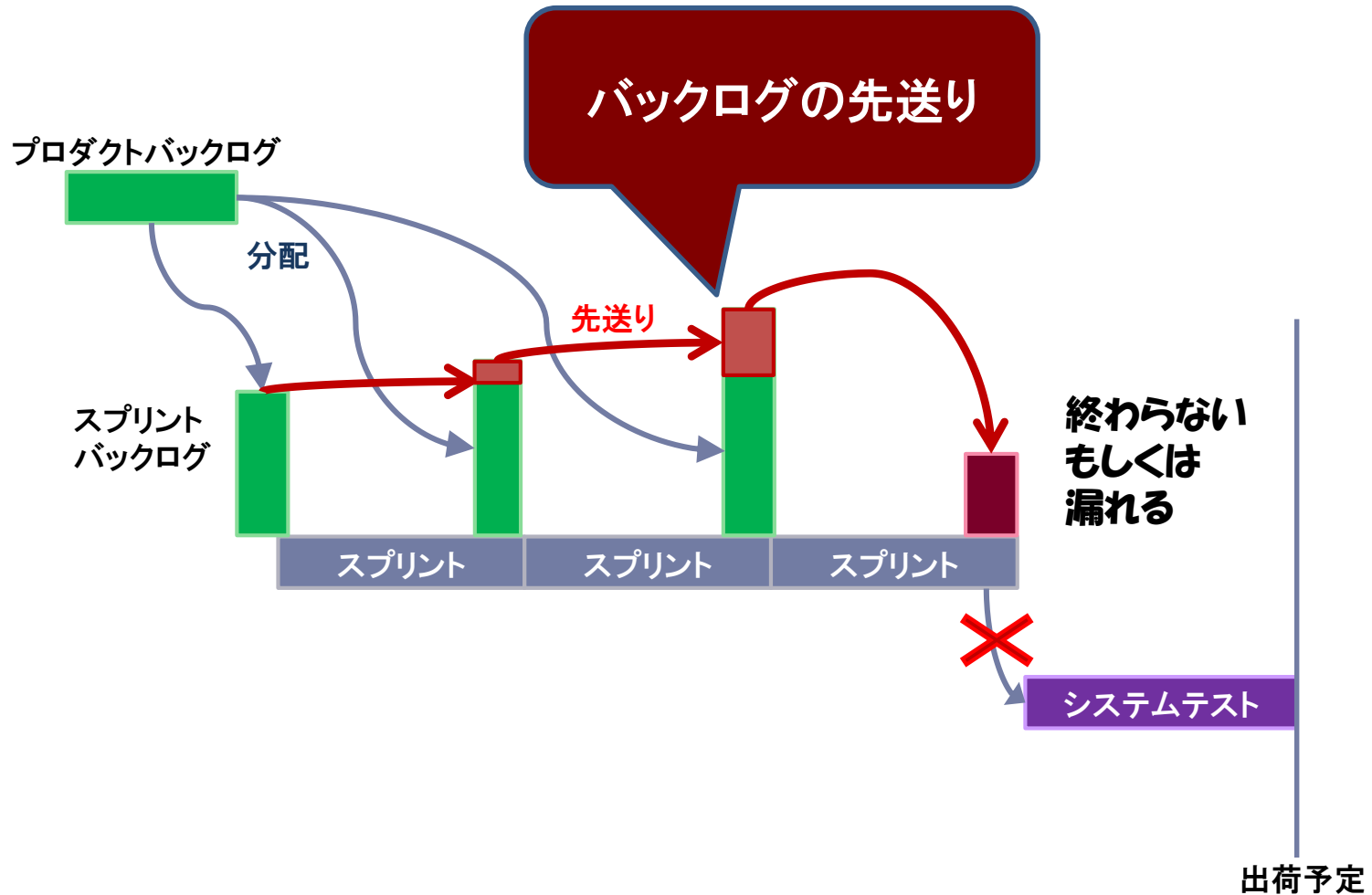


## バーンアップチャート

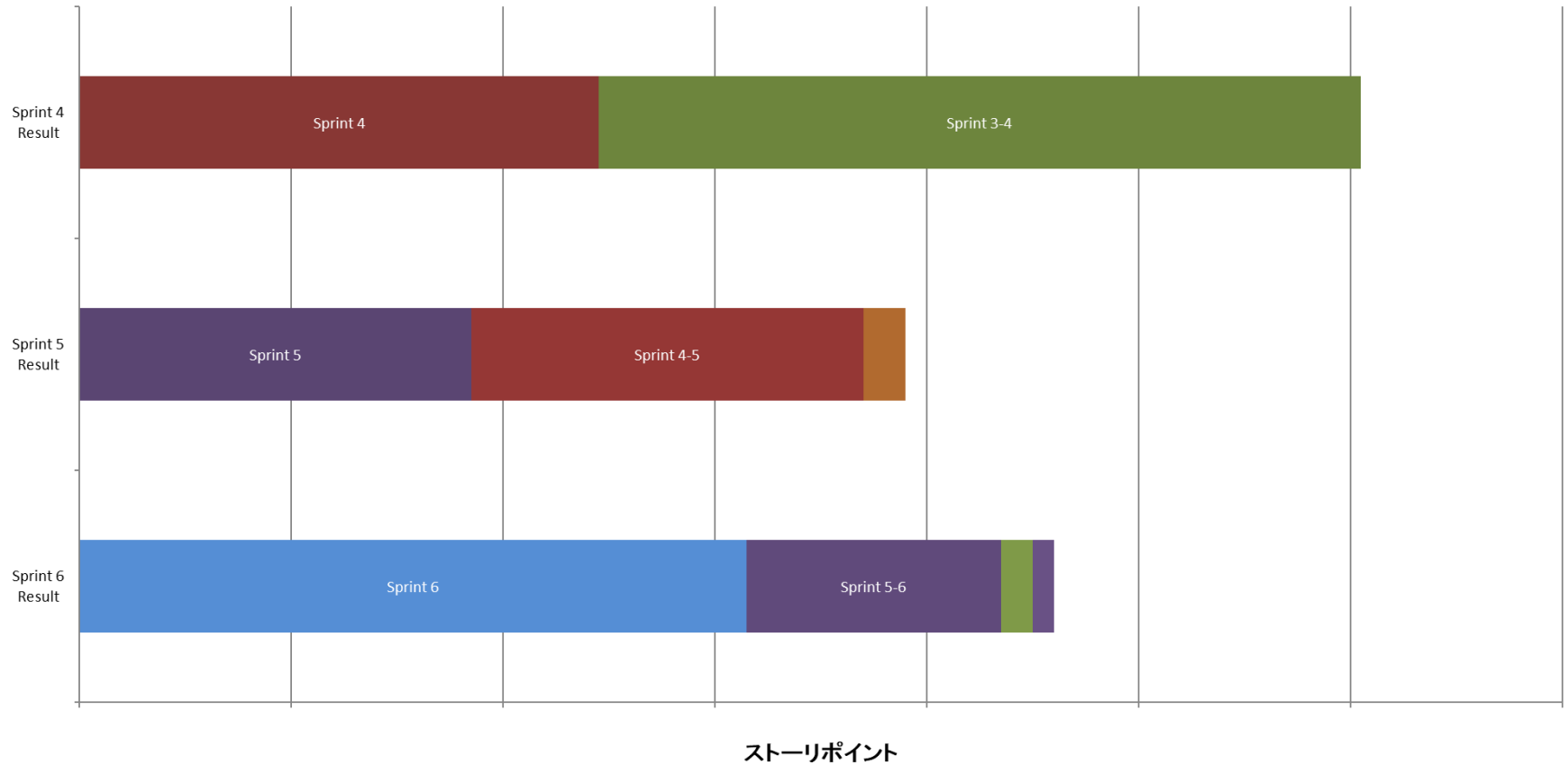
# 事例1：Velocity 以上にスプリントバックログは積めない



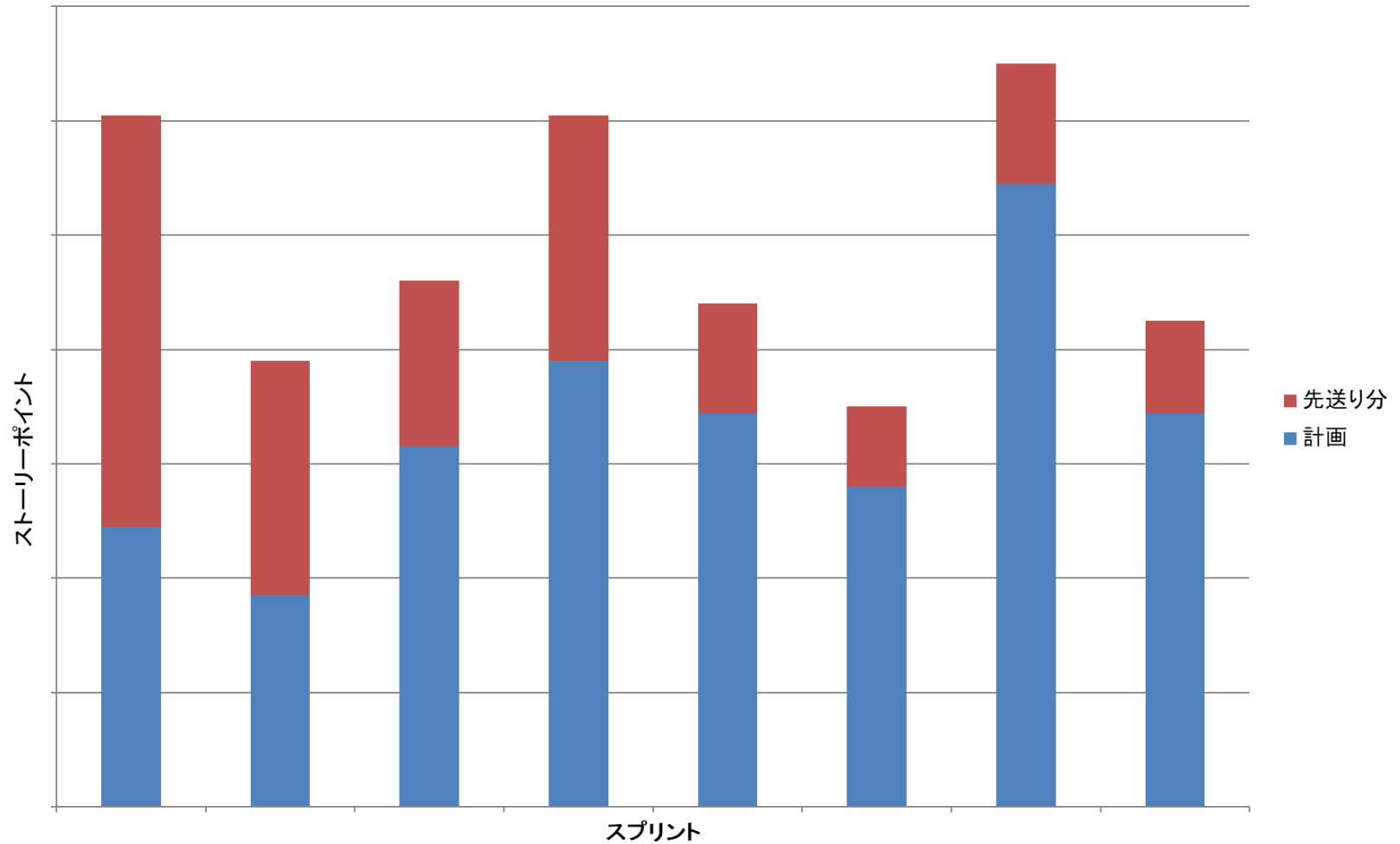
# バックログの先送り



# 先送りされているバックログ

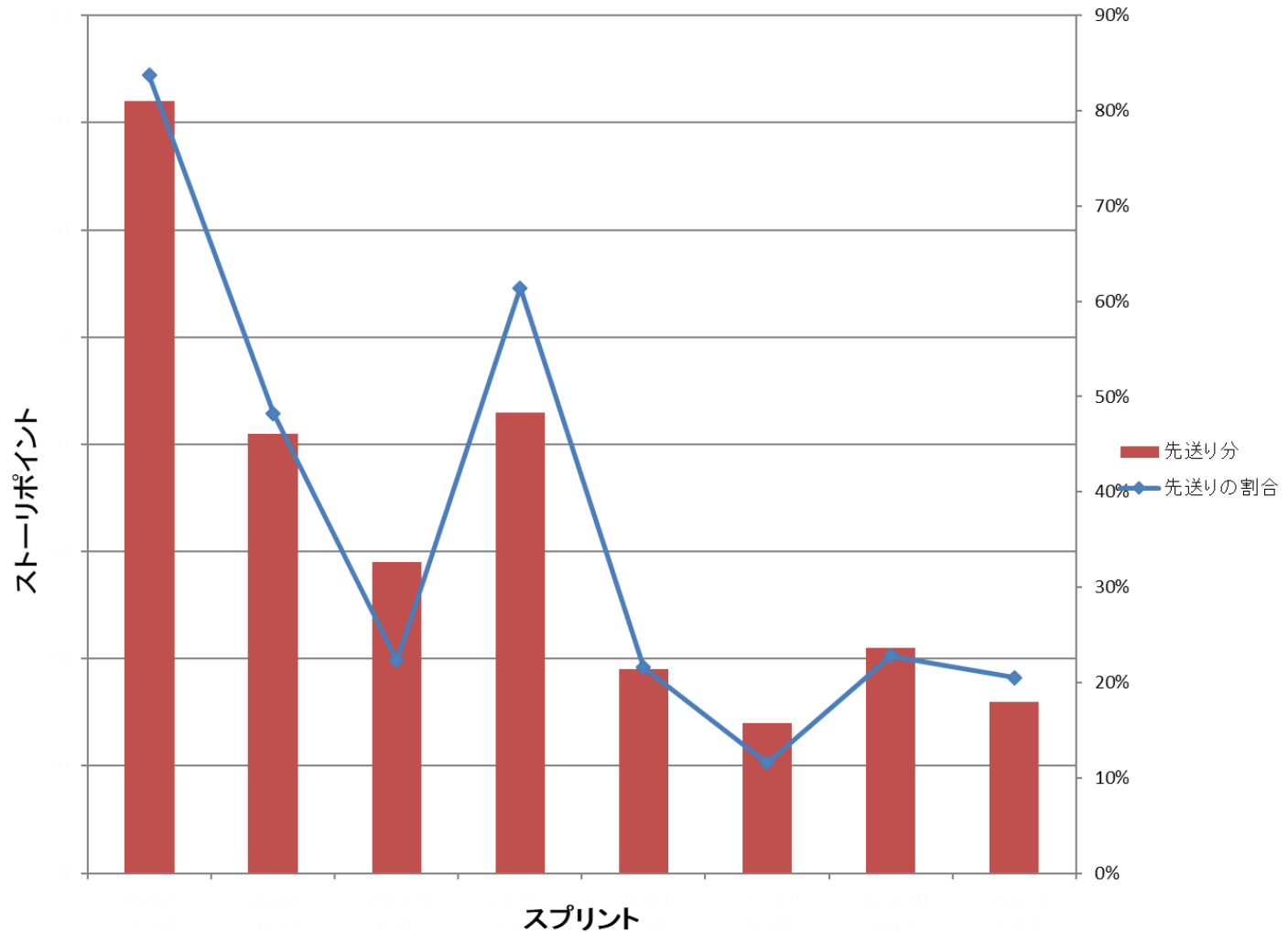


# 先送りの変化



# 先送りの減少

## スプリント毎の計画と先送り分の推移



## 事例2 背景

---

- ▶ チームメンバー 9名
  - ▶ スクラムマスタとプロダクトオーナー兼務
  - ▶ プロジェクトマネージャ（課長兼務）
  - ▶ スクラム歴 半年
- 
- ▶ 2週間スプリント
  - ▶ Sprint 11スプリント

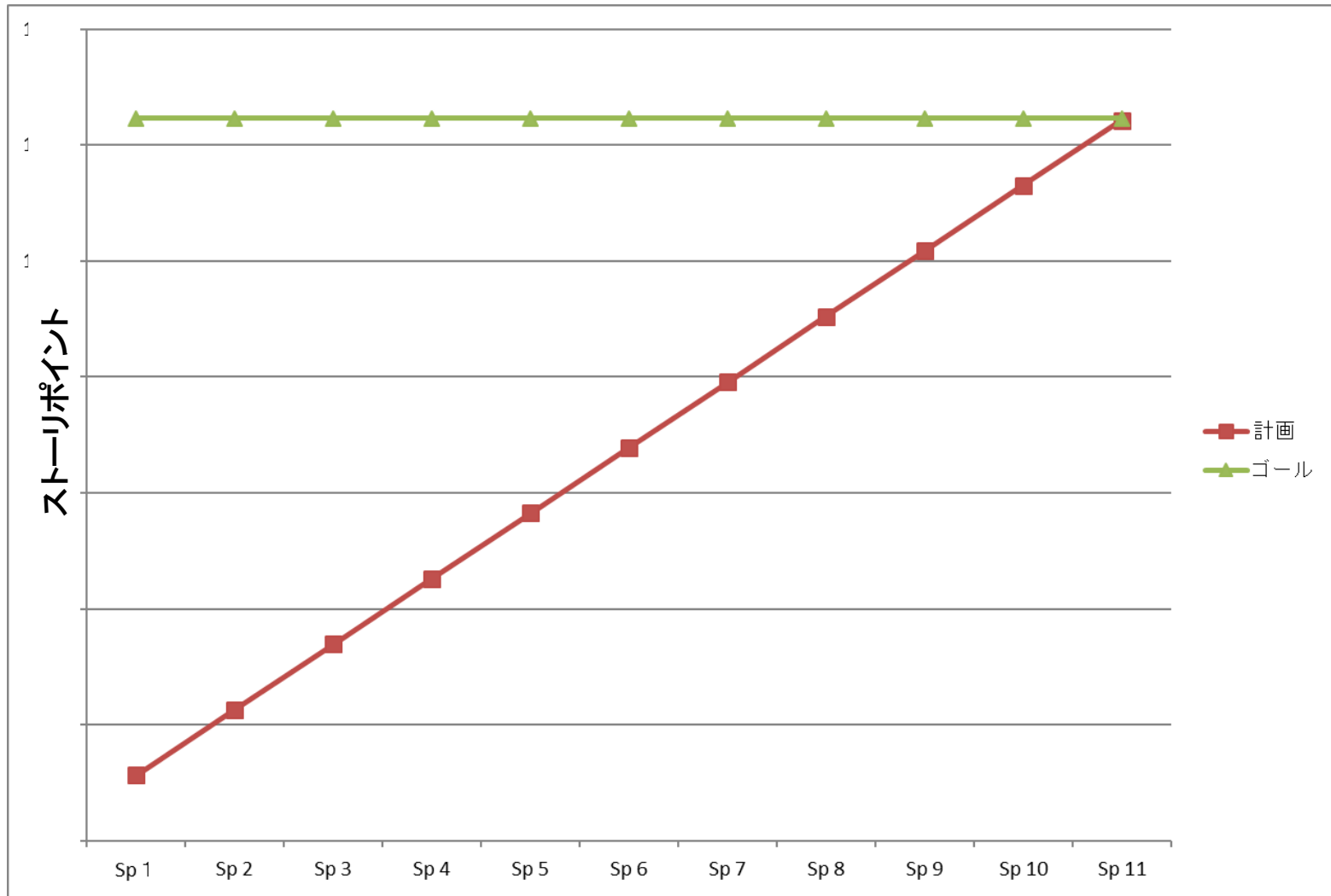
# プロダクトバックログ

---

- ▶ **製品**
  - ▶ とあるITシステム
- ▶ **スプリント0**
  - ▶ **製品**
  - ▶ PBI数 468
  - ▶ ストーリポイント累計 1246

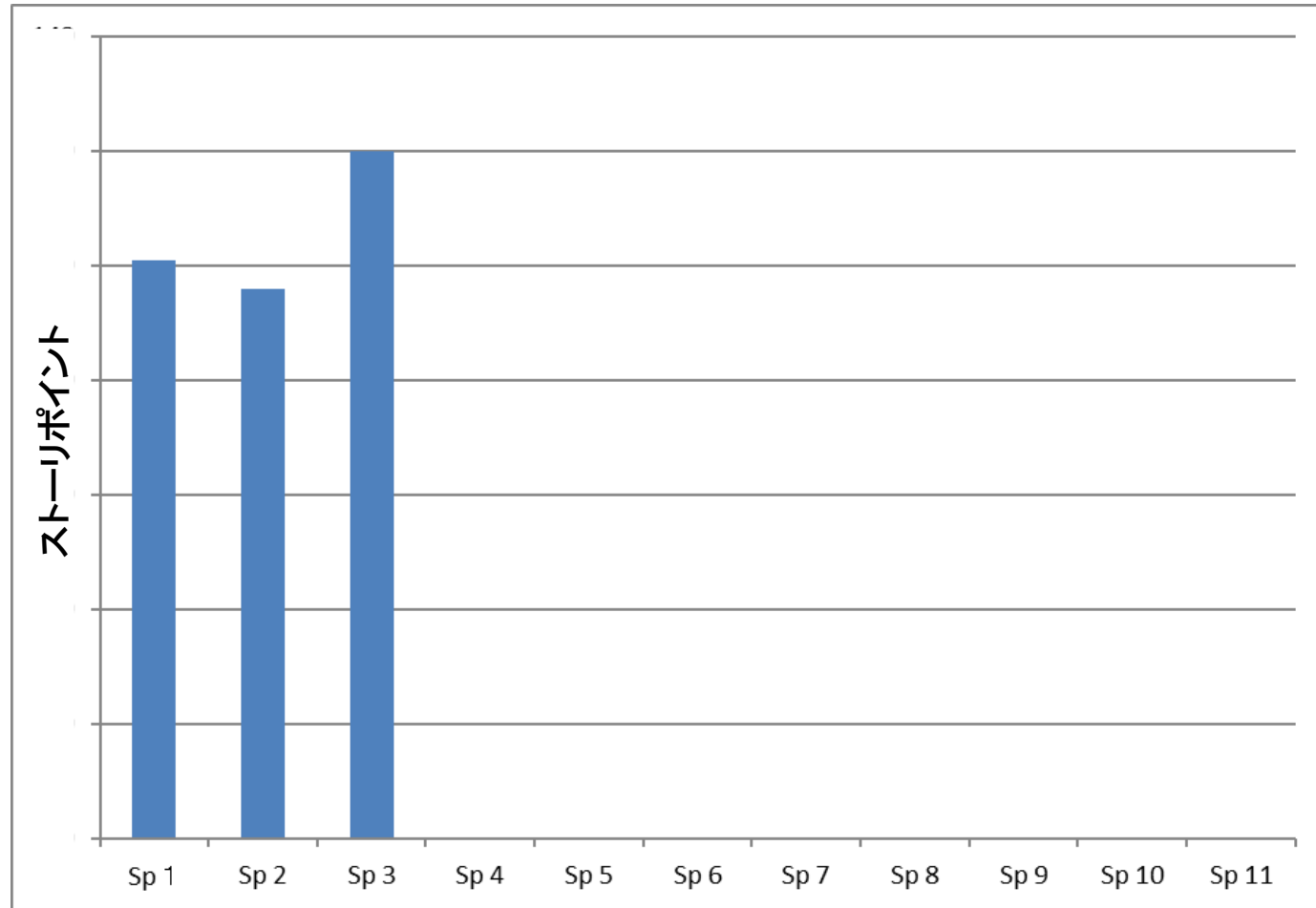


# 目論見



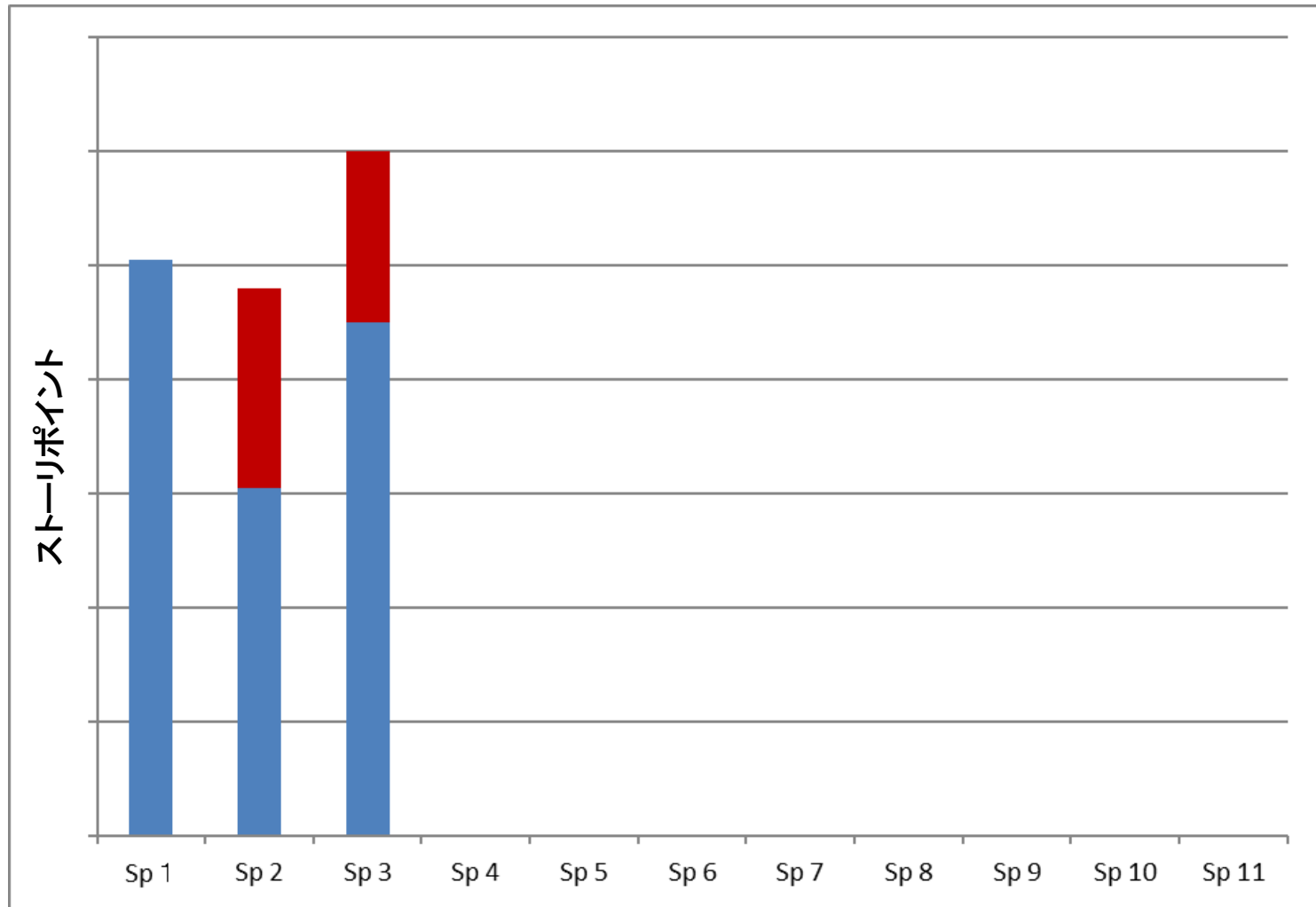
# Sprint 3 ストーリポイント累計

---

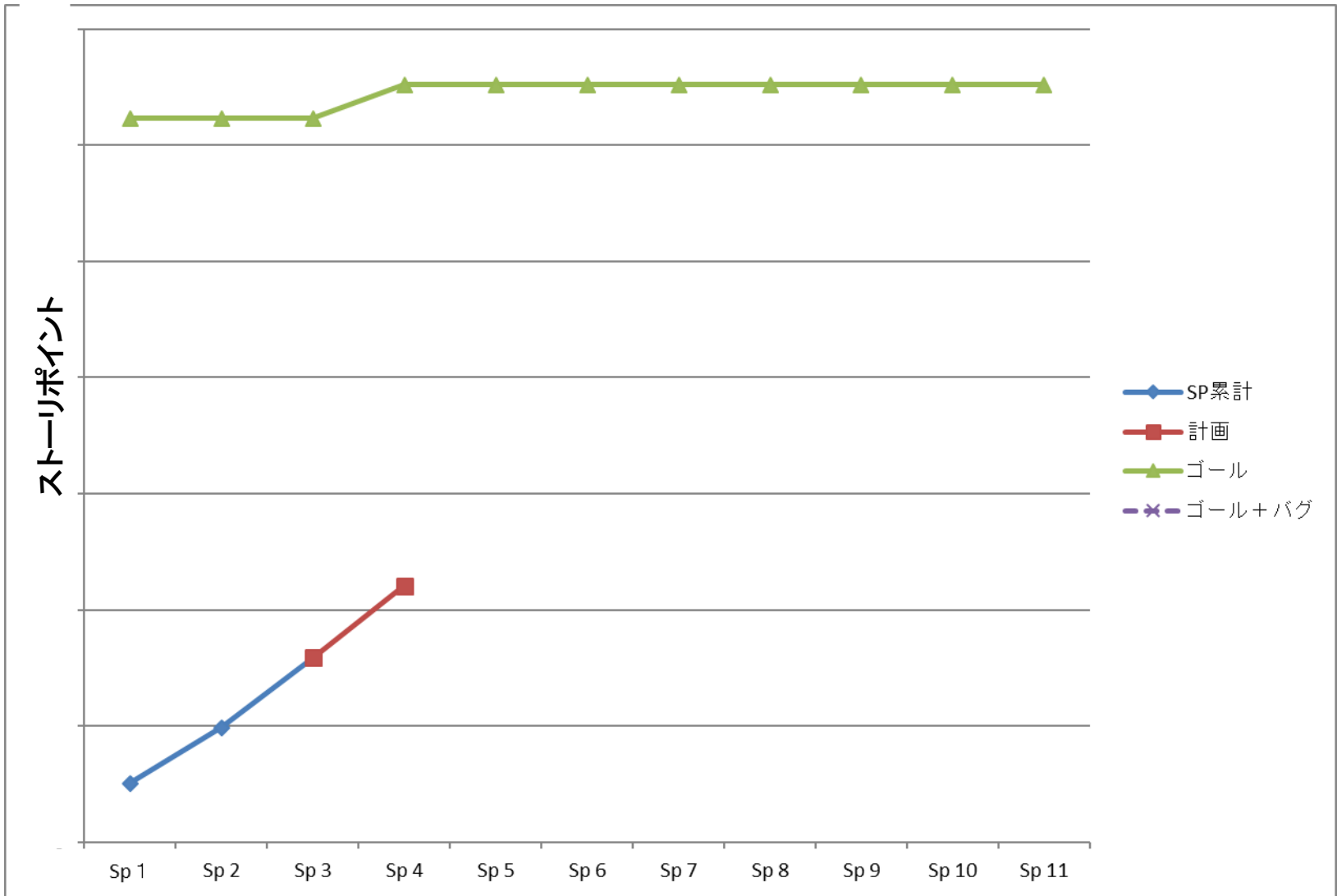


# sprint 3

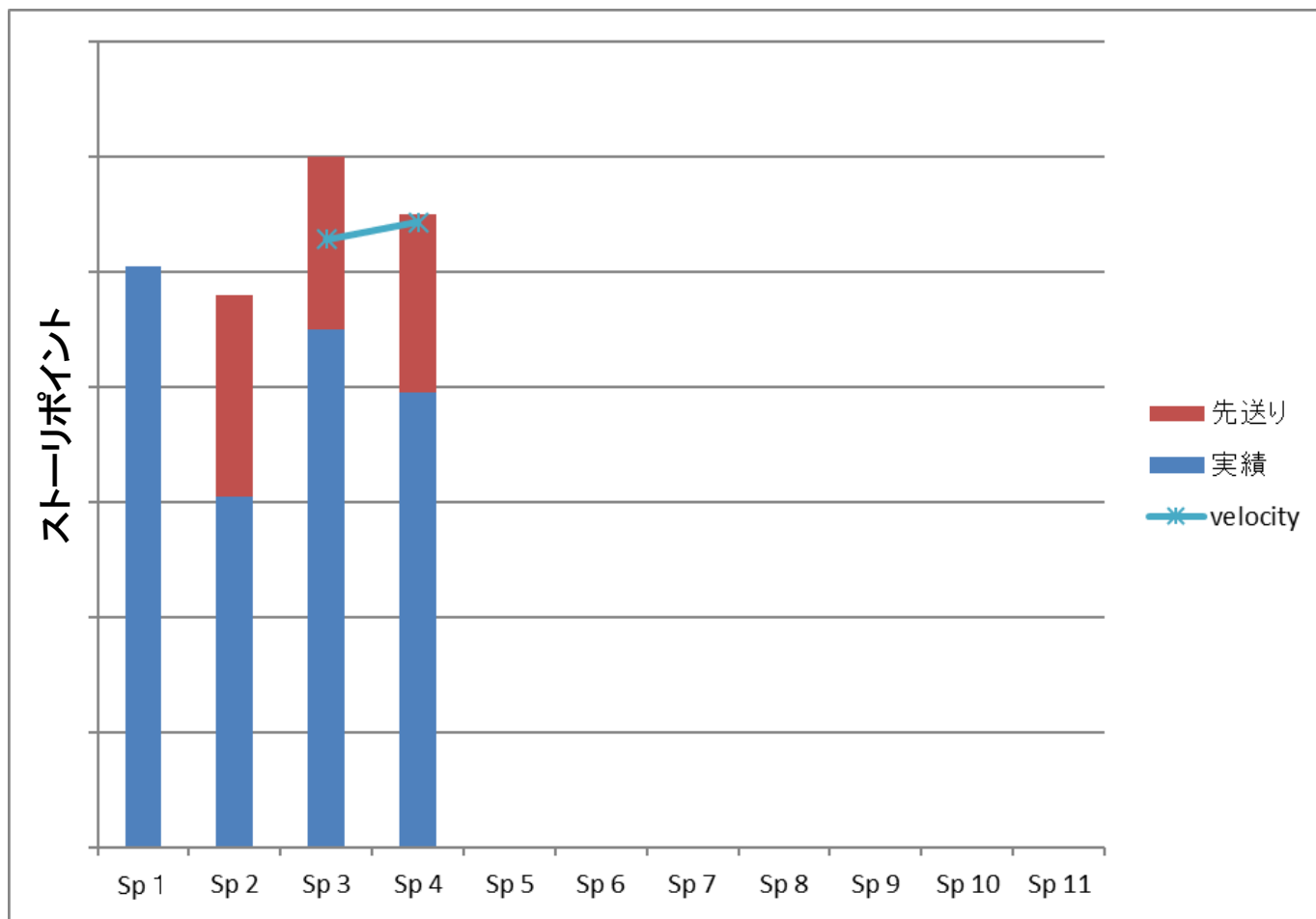
---



# Sprint 4のプランニング



# Sprint 4



## プロダクトオーナー

ある顧客へのリリースの要請

AとBとCの機能をこの時期に見せないと  
今後の営業活動ができない  
開発プロジェクト自身の存続にもインパクト  
うけざるを得ない

# Sprint 5 計画

---

**チーム**

**無理です**

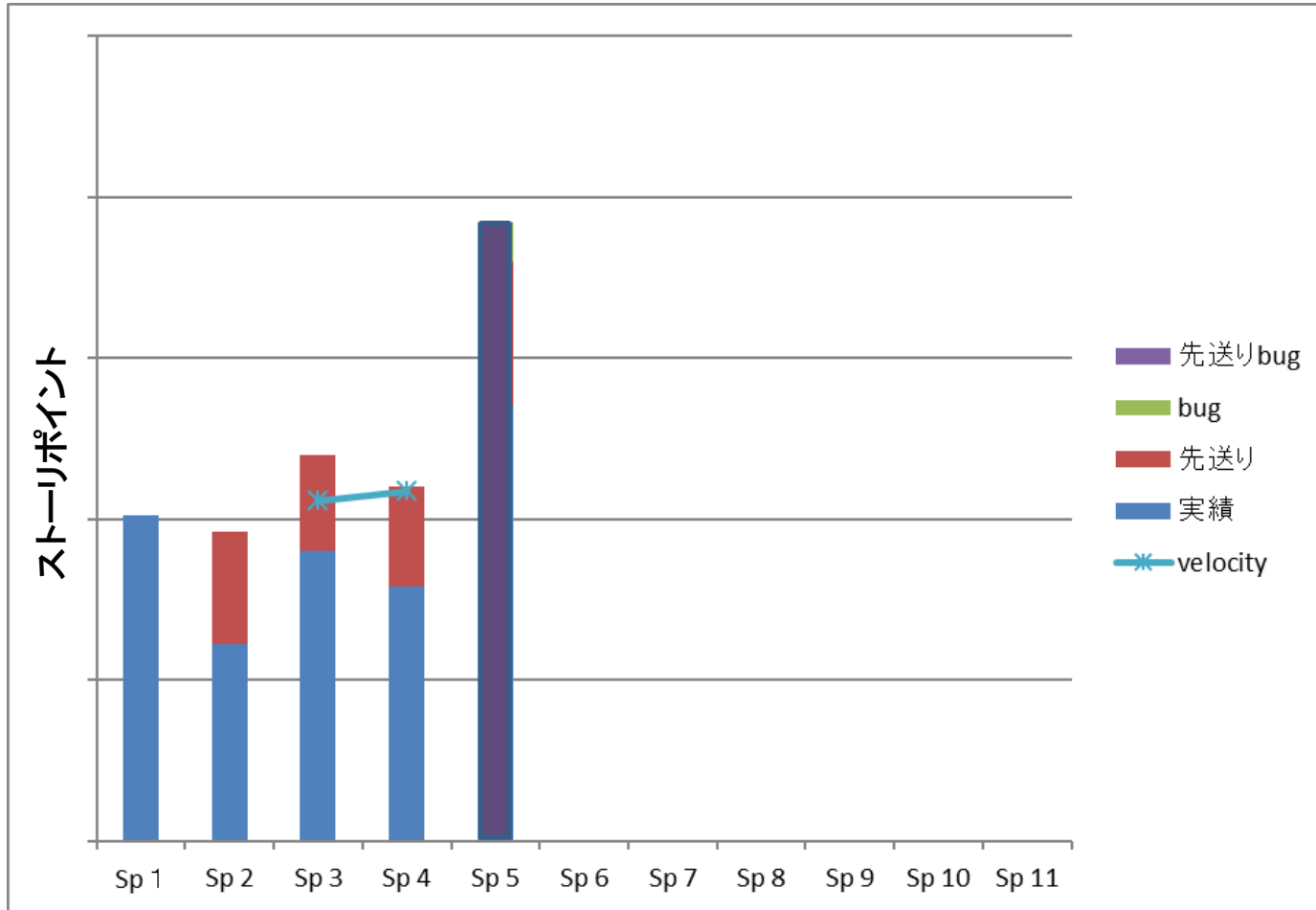
## 対策案

**ある顧客のリリースまで  
頑張ってくれないか。  
そのあとは、ペースを緩める**

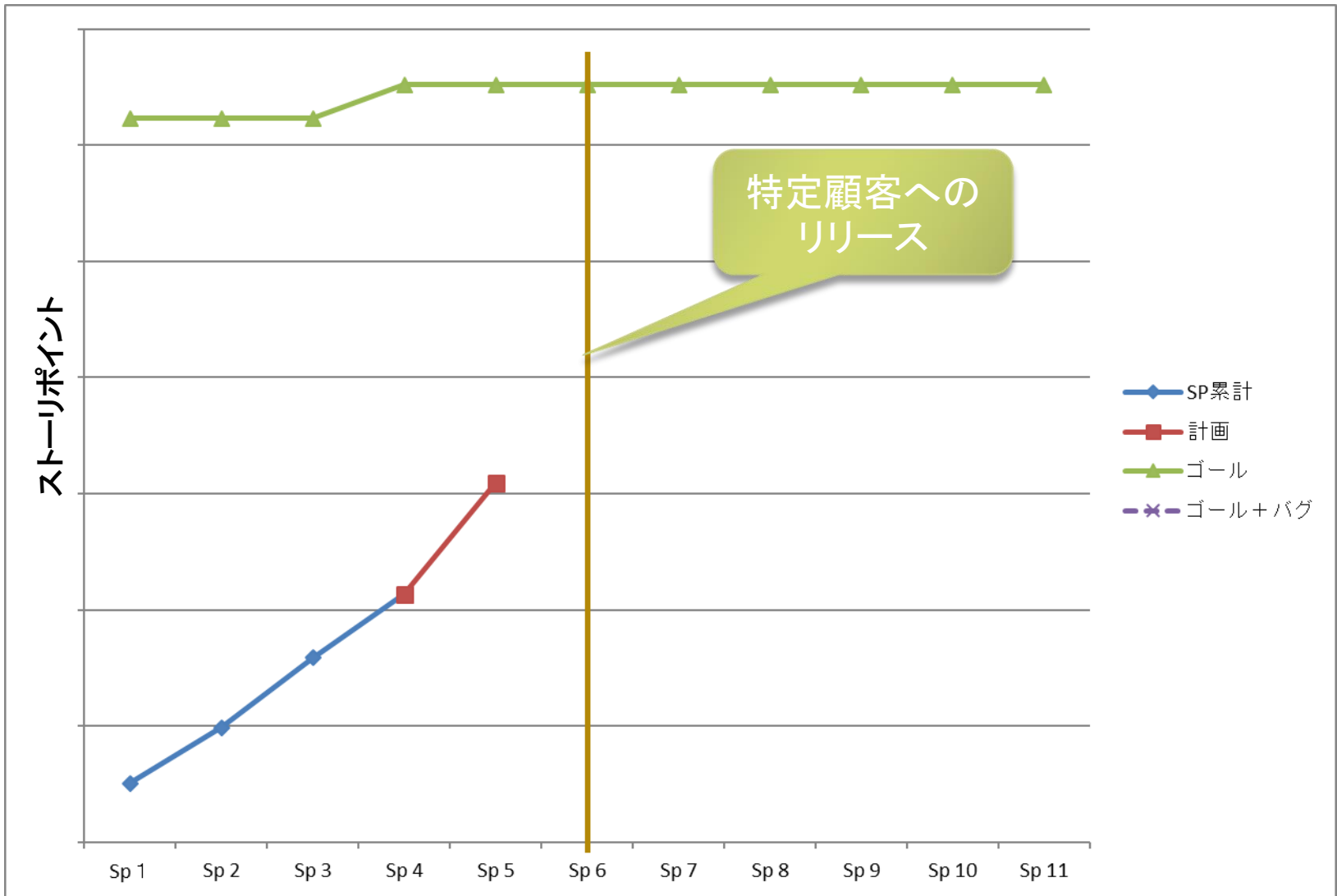
**受けざるを得ない**



# Sprint 4 計画修正後



# Sprint 5

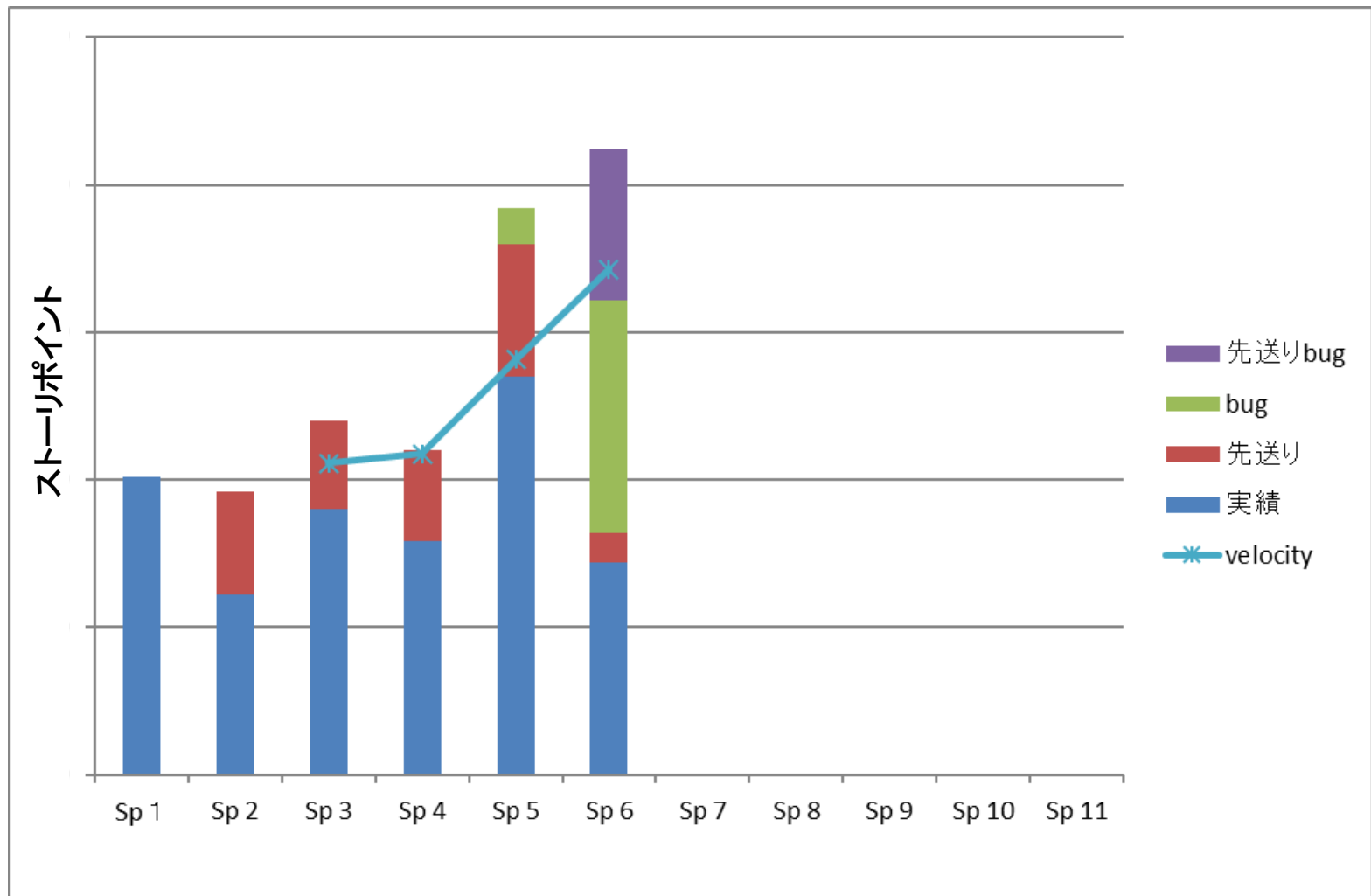


# チームには何が起こるでしょうか

---



# Sprint 5 結果

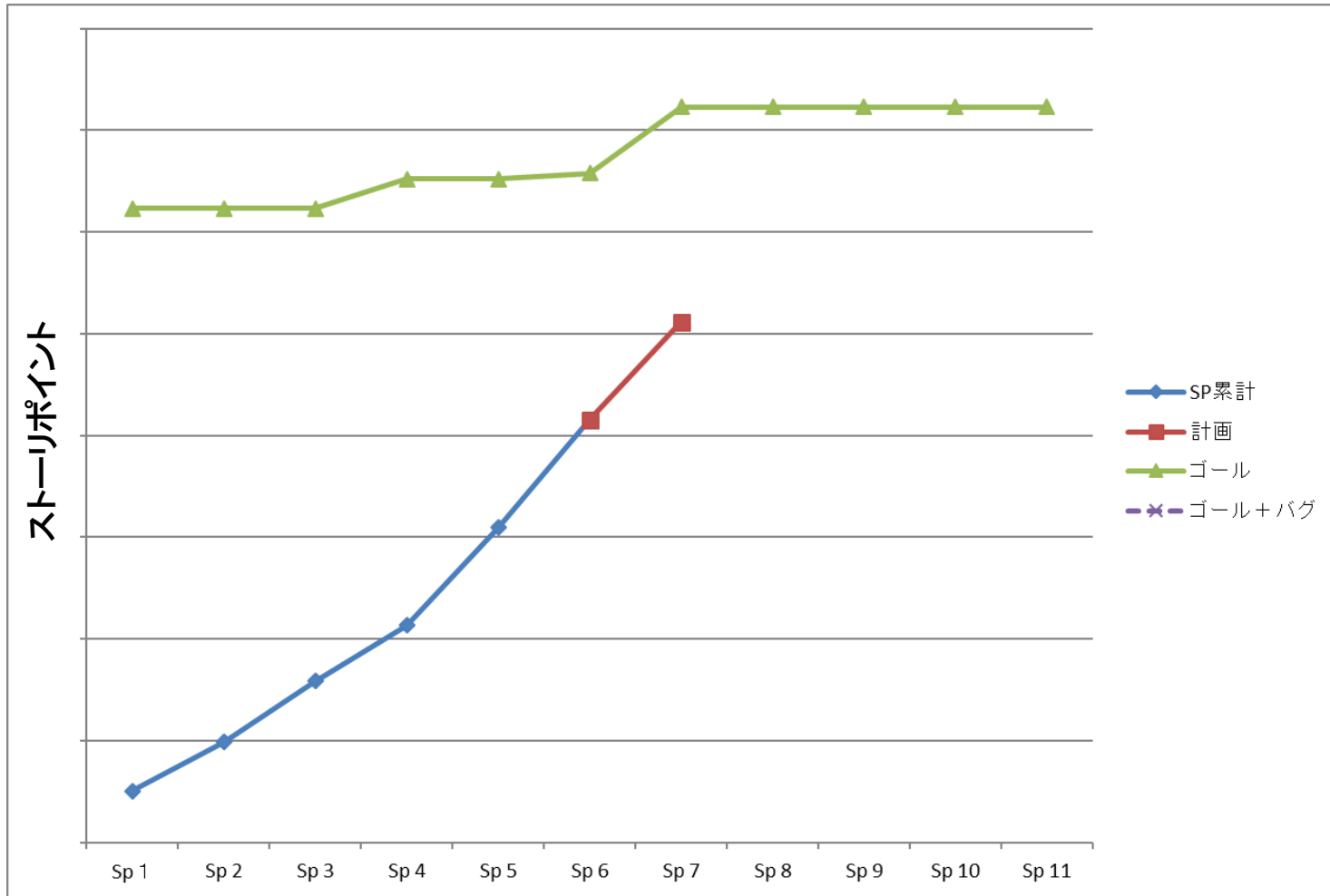


## Sprint 5で何が起きたか

---

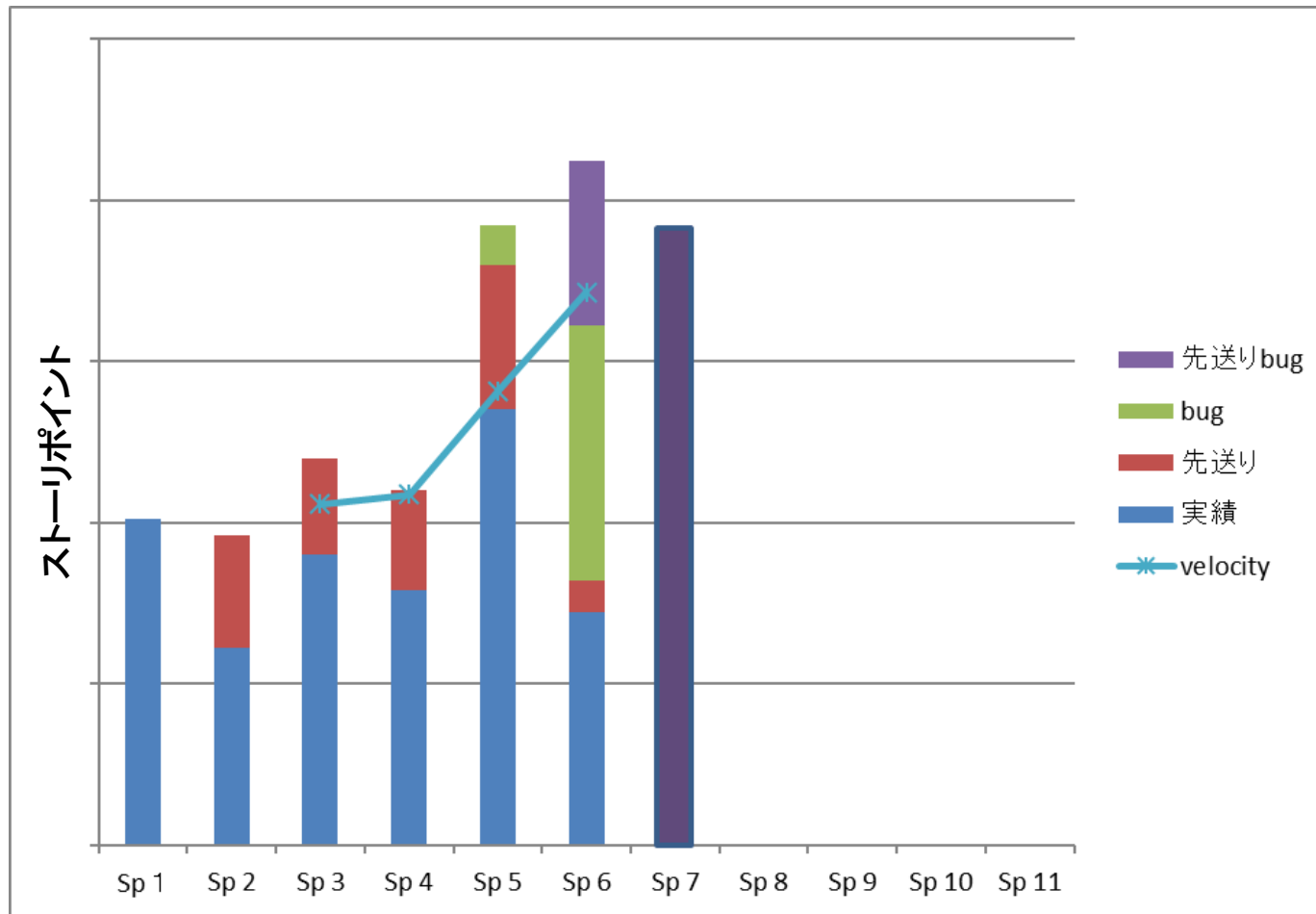
**リリースのために、システムテストが  
念入りに入ったために、  
多くのバグが発見  
それらのバグも直すために、  
少しリリースを遅らせてSP6でリリース**

# Sprint 6の計画

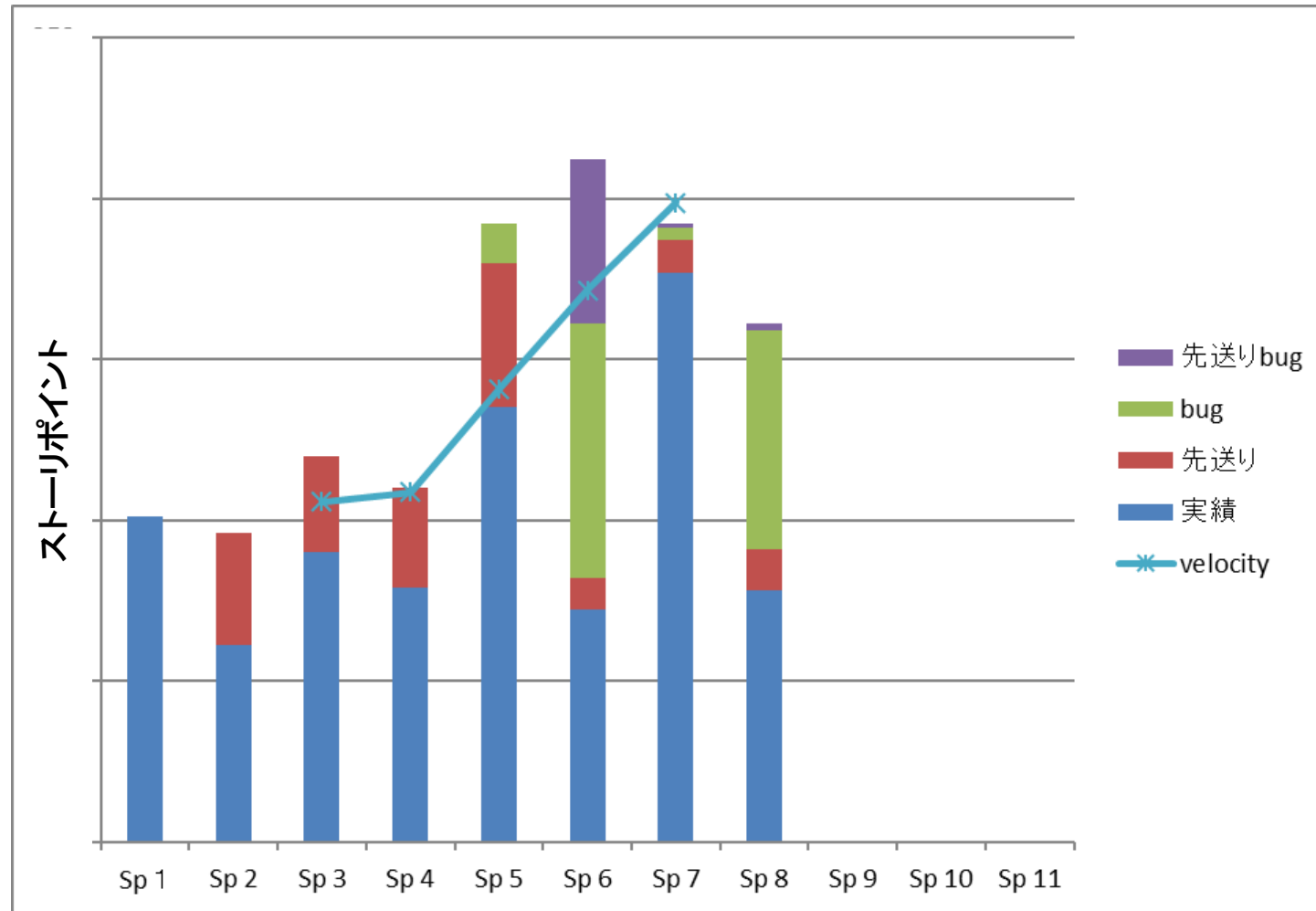


バグの分 PBIが増え、ゴールが逃げている

# Sprint 6の結果とSprint7の計画

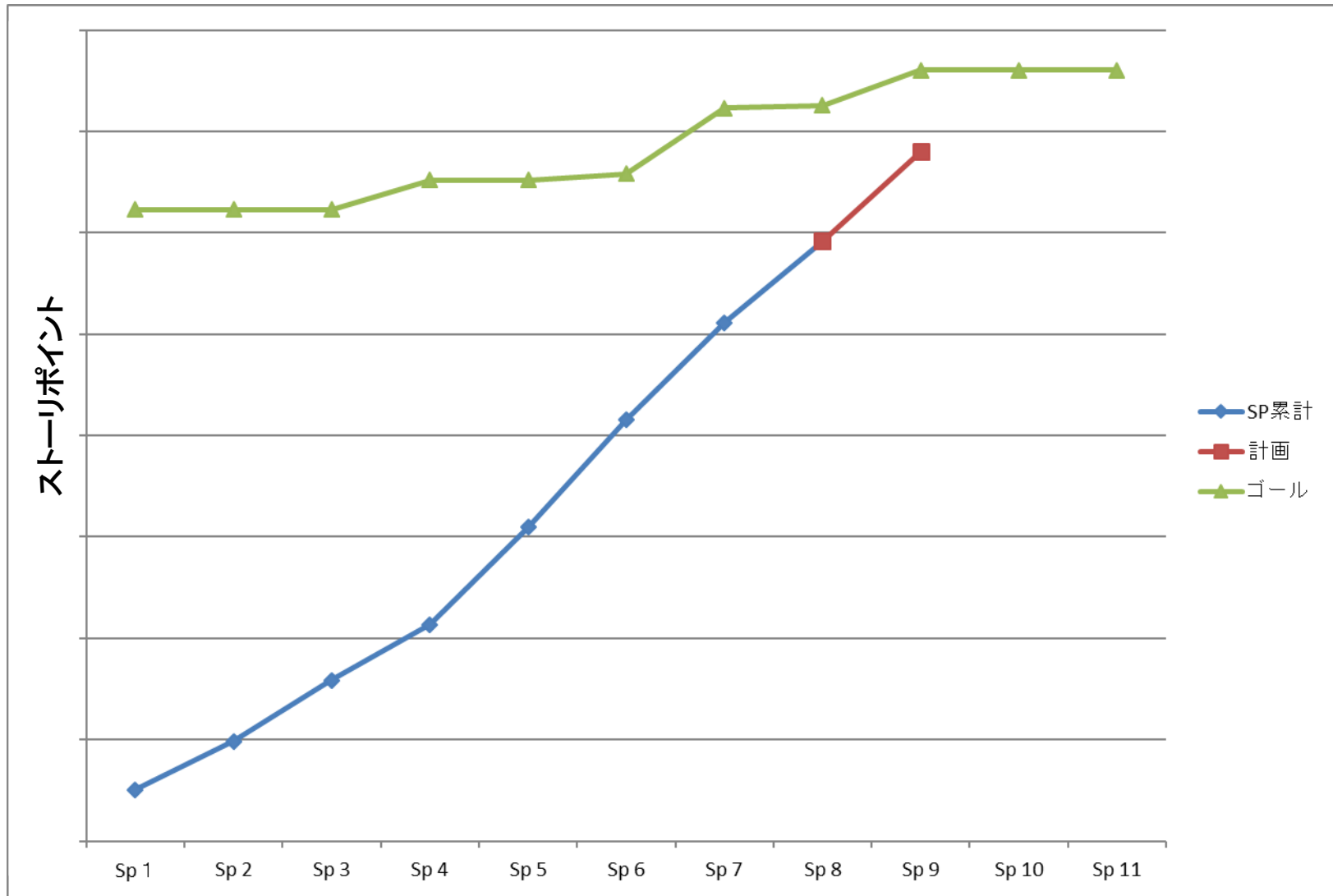


# Sprint 7の結果とSprint 8の計画

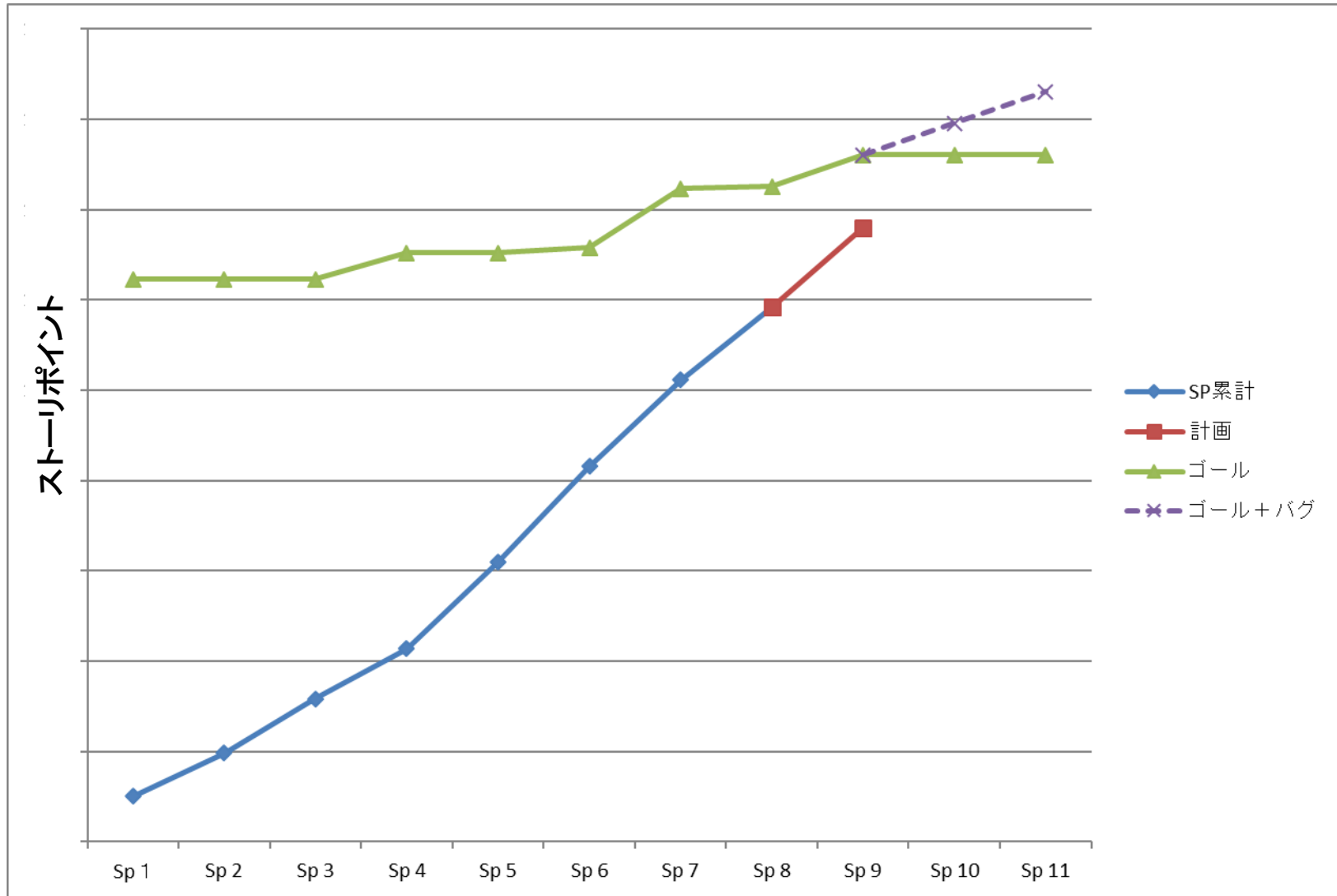




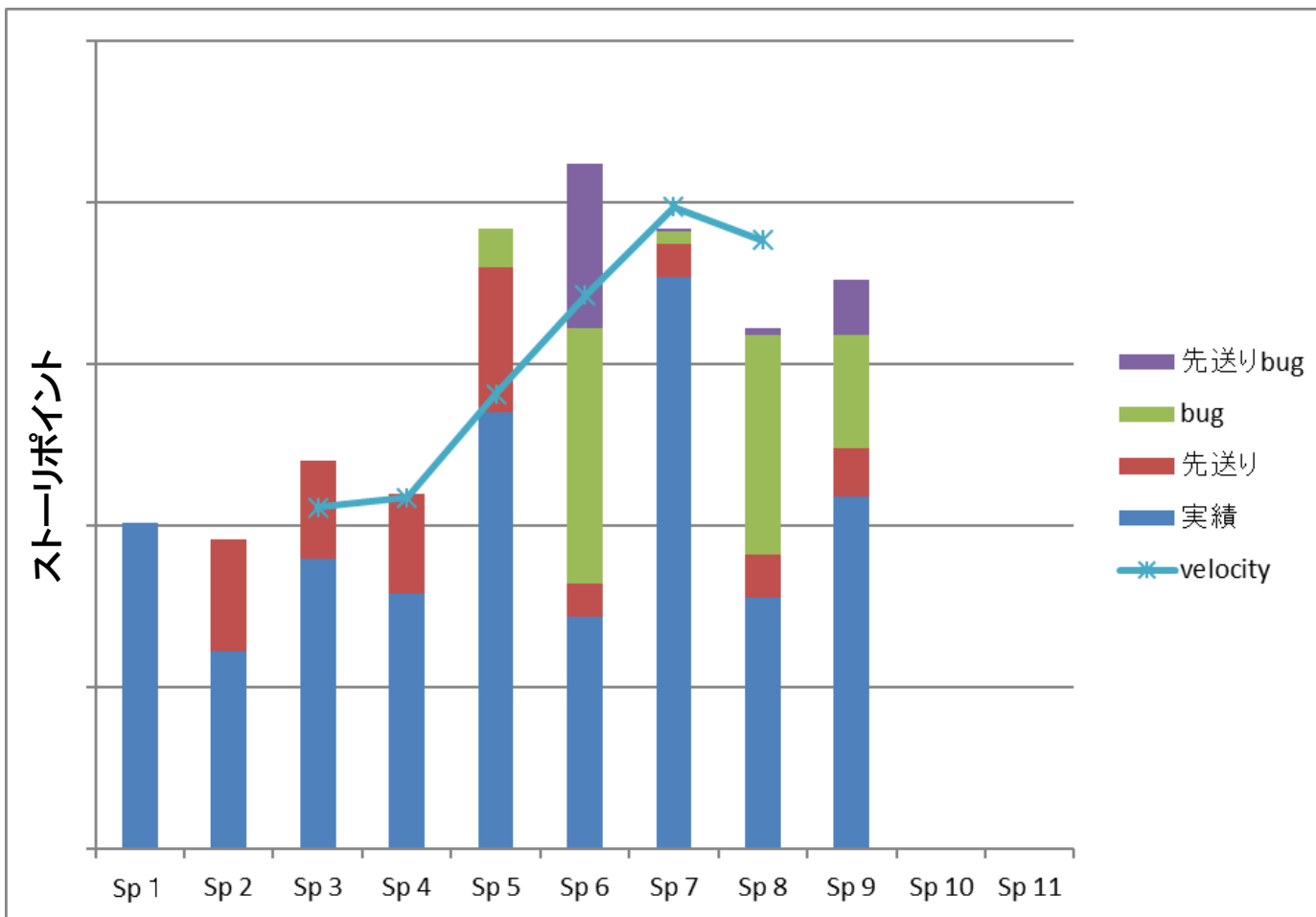
# Sprint 8の結果とSprint 9の計画



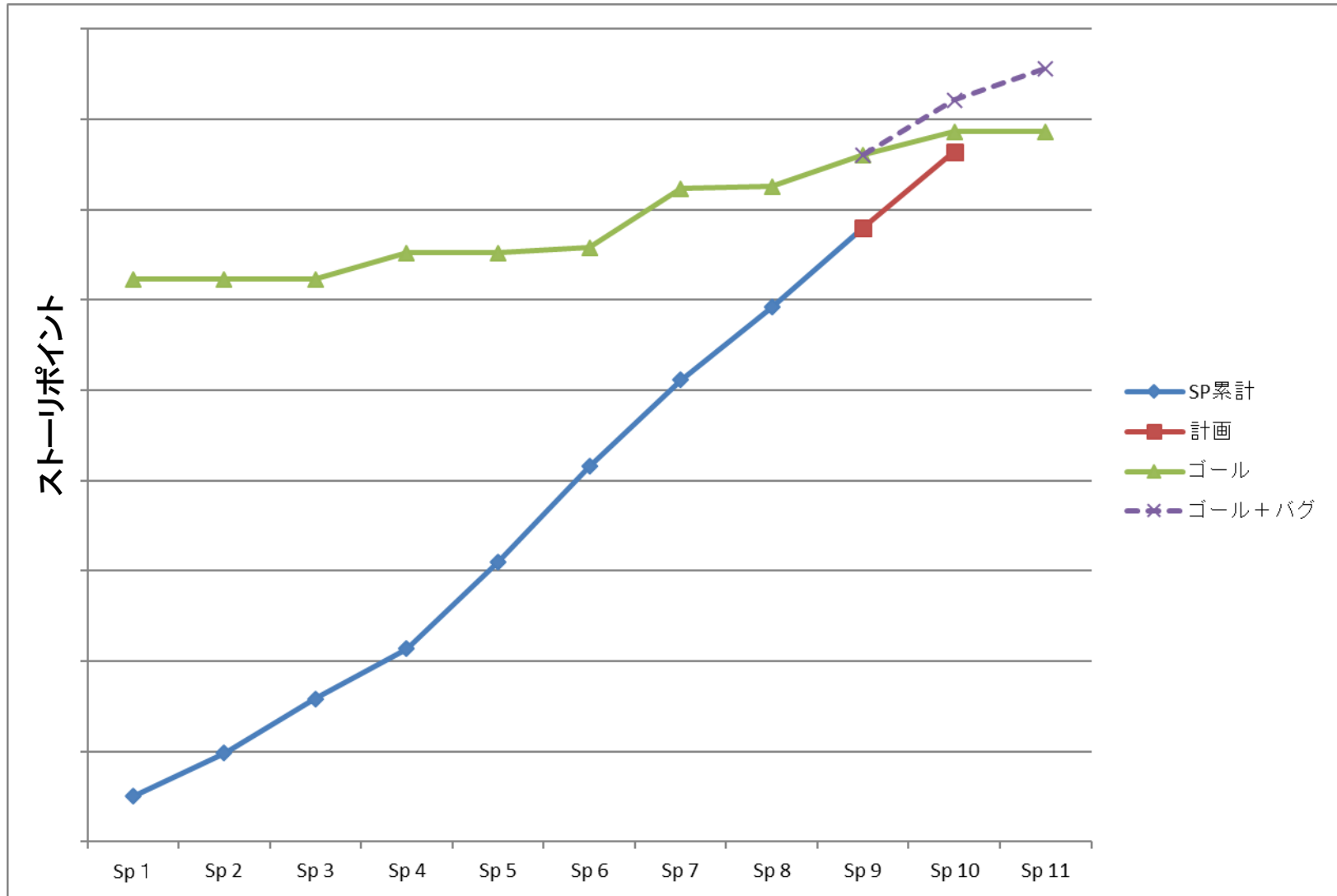
# Sprint 8の結果とSprint 9の計画



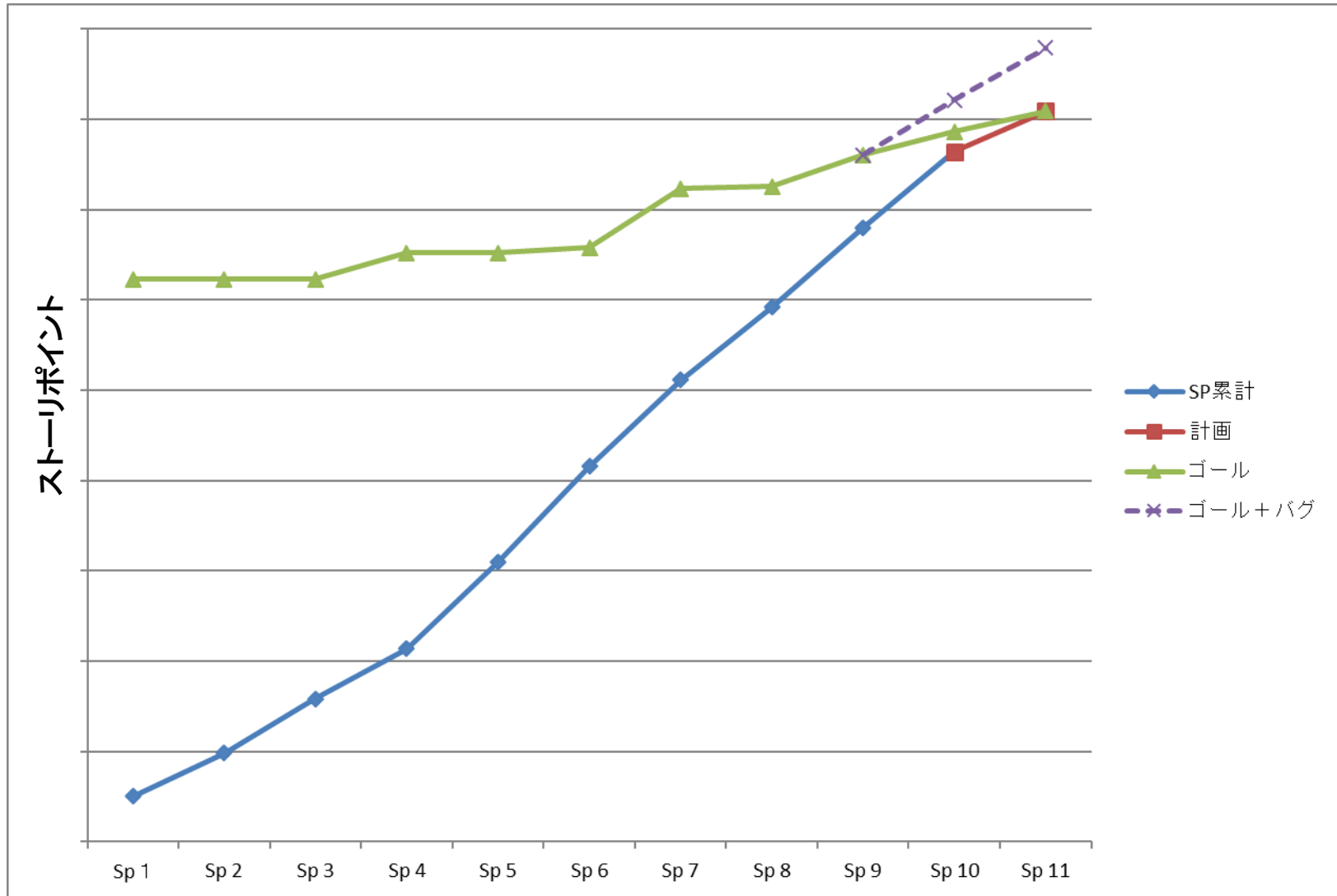
# Sprint 9計画



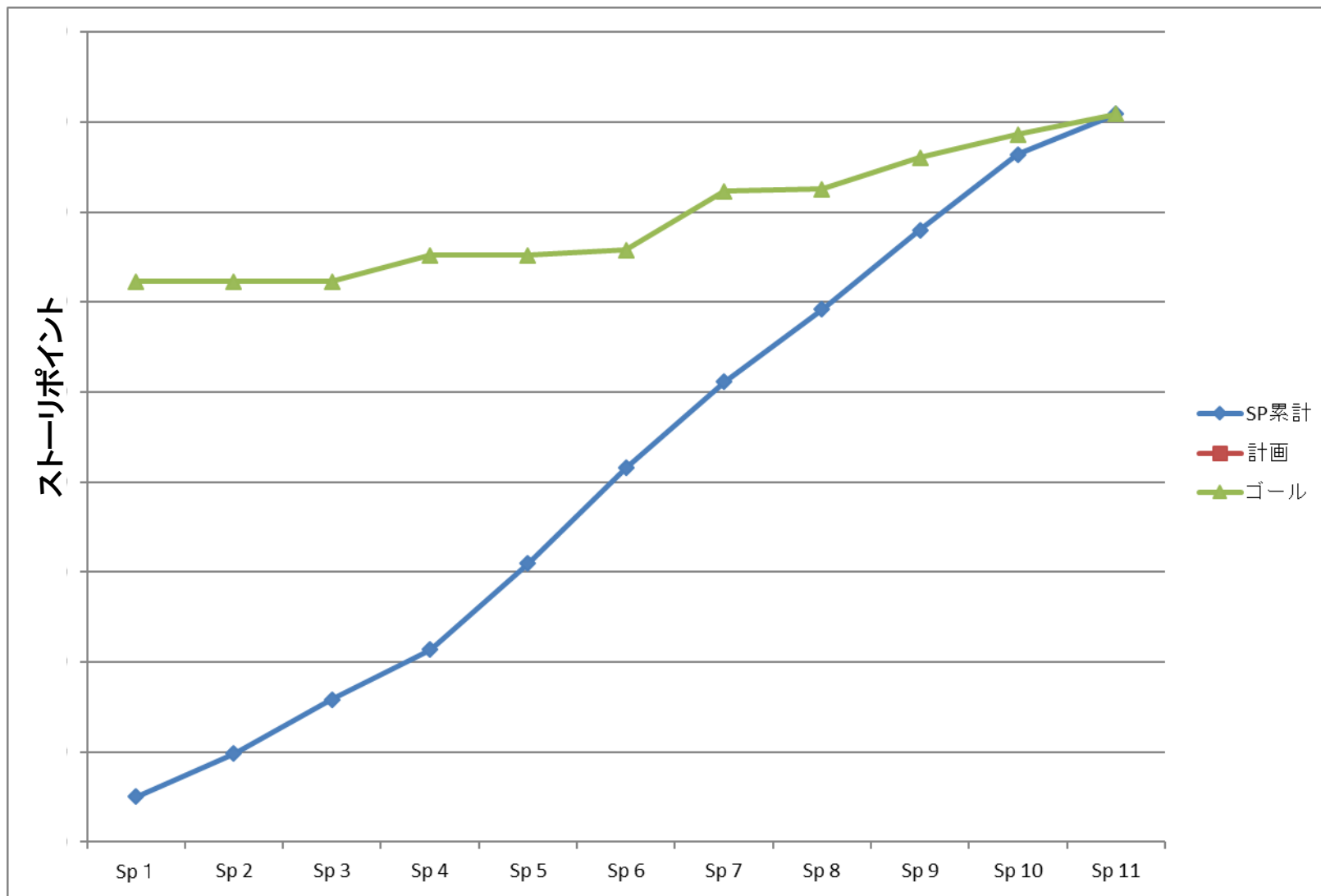
# Sprint 10 の計画



# Sprint 10の結果とSprint 11の計画



# Spint 11結果



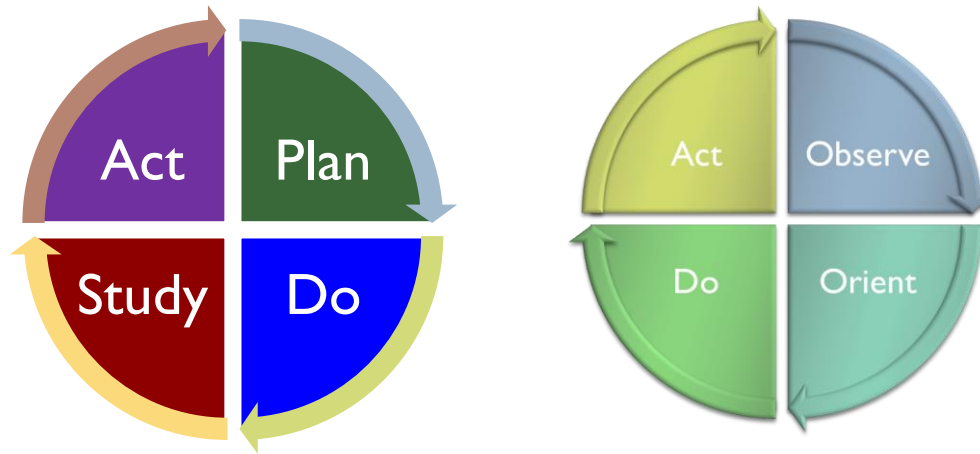
# ここから何が学べるでしょうか

---

**メトリクスのフィードバックから、  
チームはリアルタイムに適切な対処  
を行い、ゴールを達成することがで  
きた。**

# まとめ

---



メトリクスは行動を起こさせる  
メッセージを持たなければ  
ならない。

**Quality On Time**



**ありがとうございました**

