

# チーム

---

- 4～5人で1チームを作ってください。
- チームを作ったら、1名（5名チームの場合は2名）をスクラムマスター役に任命してください。
- 残りの3名は「開発者」役です。

# 自己紹介 – 松永広明



松永広明

- ◆ 株式会社 豆蔵 シニアコンサルタント(2015年～)
- ◆ 1990年 株式会社安川電機
  - 産業用インバータ事業に従事
- ◆ 1997～2015年までソニー株式会社
  - デジタル携帯電話、デジタルイメージング事業などに従事
- ◆ 2011年ごろよりアジャイル開発を推進。
- ◆ 書籍
  - わかりやすいアジャイル開発の教科書(2012) レビューアー
- ◆ 資格：Scrum Alliance Certified Scrum Professional
- ◆ 講演：Scrum Gathering Tokyo 2016  
ESEC2016, ET/IoT2016, など



# シン・コラボレーション

---

～ワークショップで体感するコラボレーションの"深"化～

コラボレーション、してますか？

# コラボレーションとは

Collaboration

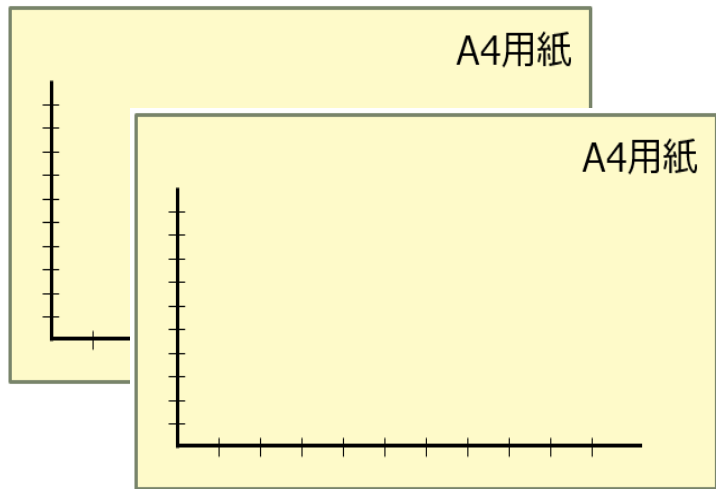
【名】協力、協調、協業、共同制作（作品）  
協同研究



演習

# 準備

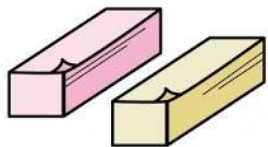
□ 下記の本物が揃っていることを確認してください。



A3用紙

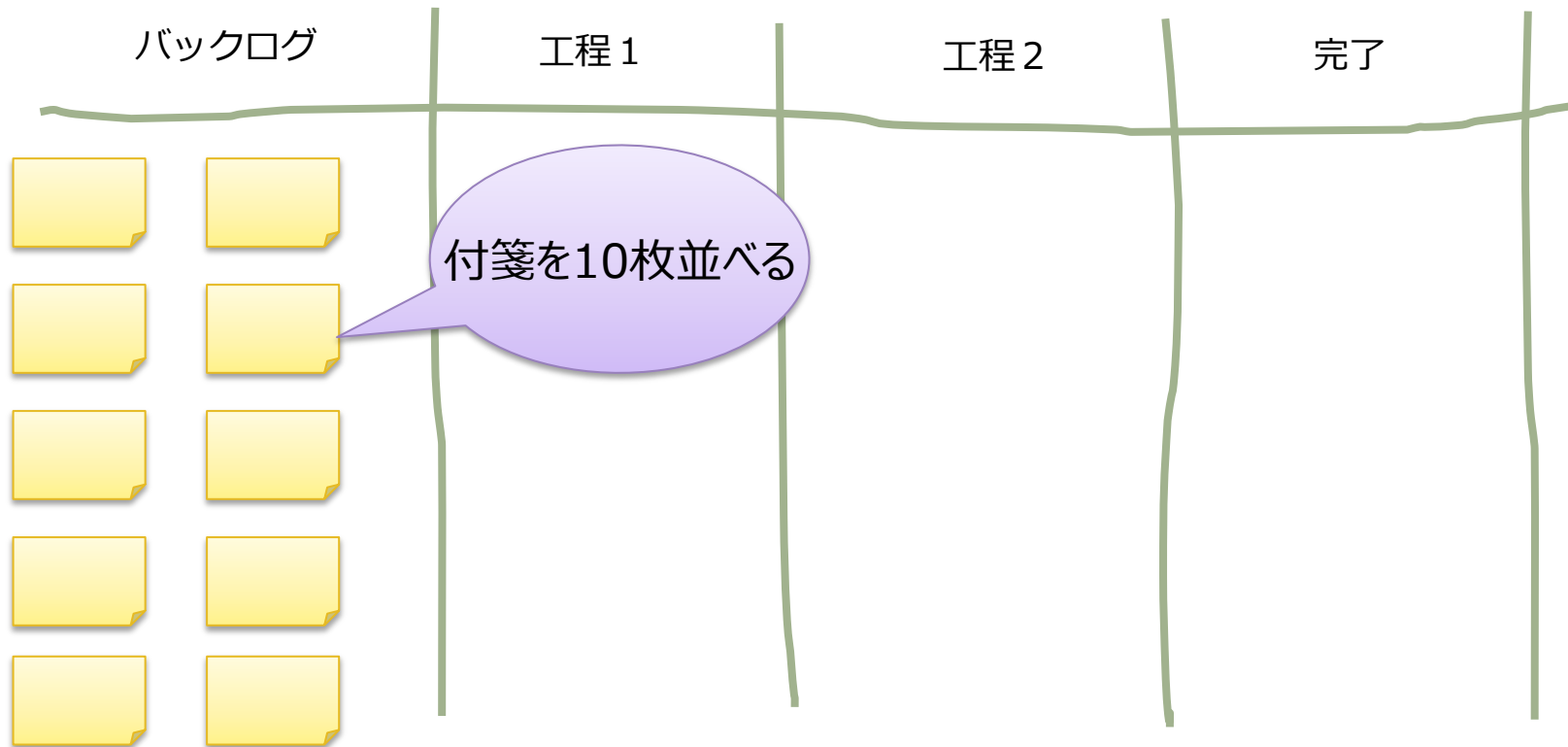
バックログ	工程1	工程2	完了

The image shows a large A3 sheet of paper divided into four columns by vertical lines. The columns are labeled from left to right: 'バックログ' (Backlog), '工程1' (Process 1), '工程2' (Process 2), and '完了' (Completed). A horizontal line is drawn across the top of the columns, just below the labels.



# タスクボード初期化

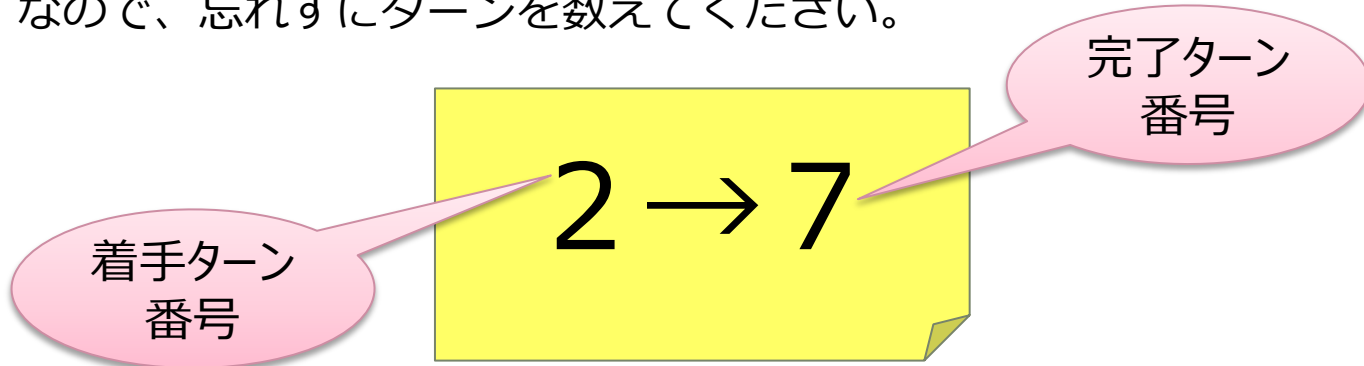
## □ 初期状態





# 基本ルール

- 開発者3名で実施します。
- 3名全員が、ひとり一回ずつ付箋紙を動かすと、それで1ターンとします。
  - ◆ つまり、1ターンにひとり一回付箋を動かします。
  - ◆ 一回に動かせるのは、一列分だけです。
- 1つのターンで同じ付箋紙を二回以上動かすことは出来ません。
- スクラムマスターは、付箋紙に着手した（=工程1に入った）ターン番号と、完了したターン番号を記入してください。
  - ◆ なので、忘れずにターンを数えてください。



# 演習 1 進め方

---

- 付箋をバックログに10枚貼ってください。
- 10枚全部の付箋を順次工程 1 に移動させてください。
- それが終わったら、10枚全部の付箋を順次工程 2 に移動させてください。
- それが終わったら、10枚全部の付箋を順次「完了」に移動させてください。
  - ◆ 1 ターンに移動出来る付箋は、一人一枚です。
- スクラムマスターは、各ターンで「完了」**になっていない**付箋の数を数えて、バーンダウンチャートを描いてください。

# 演習 1

---

□ではスタート！！

# 演習 1 結果確認

---

- 全部の付箋が「完了」になるのに、何ターンかかりましたか？
- 各付箋紙の、着手から「完了」までのターン数は平均いくつでしたか？

どこかに数字を  
記録してください。

## 演習 2 進め方

---

- 演習 1 の付箋を全部剥がして、新しい付箋をタスクボードにセットしてください。
  - ◆ バックログに付箋紙10枚。
- **今回は、工程 1 および工程 2 には、付箋紙を最大でそれぞれ 2 枚までしか置いてはいけません。**
- それ以外は演習 1 と同様に進めます。
  - ◆ 1 ターンに動かせるのは、1 列分だけです。
  - ◆ 1 つのターンで同じ付箋紙を二回以上動かすことは出来ません。

## 演習 2

---

□ではスタート！！

## 演習 2 結果確認

---

- 全部の付箋が「完了」になるのに、何ターンかかりましたか？
- 各付箋紙の、着手から「完了」までのターン数は平均いくつでしたか？

どこかに数字を  
記録してください。

# 結果比較

---

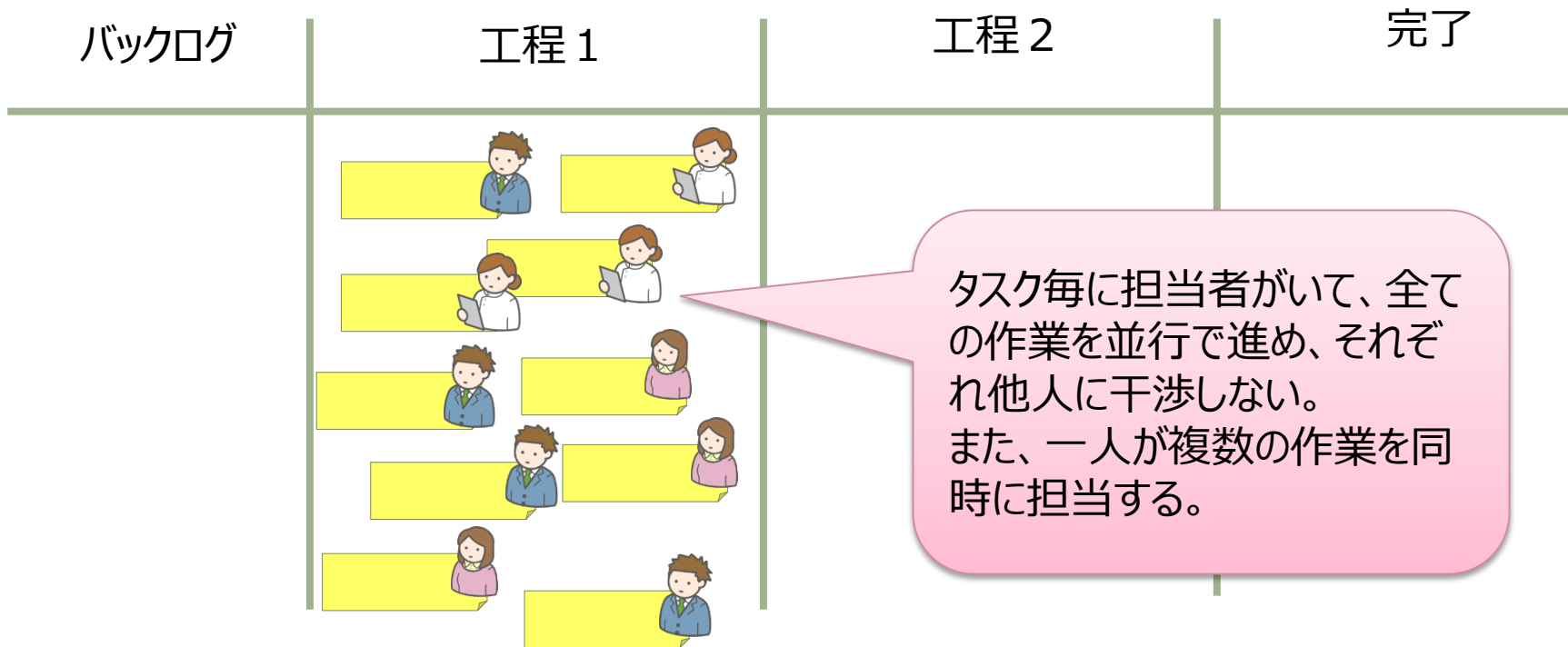
□ 演習1と演習2の結果を比較して、どうでしたか？

- ◆ 各付箋紙の、着手から「完了」までのターン数
- ◆ 全部の付箋が「完了」になるまでのターン数
- ◆ バーンダウンチャートの形状



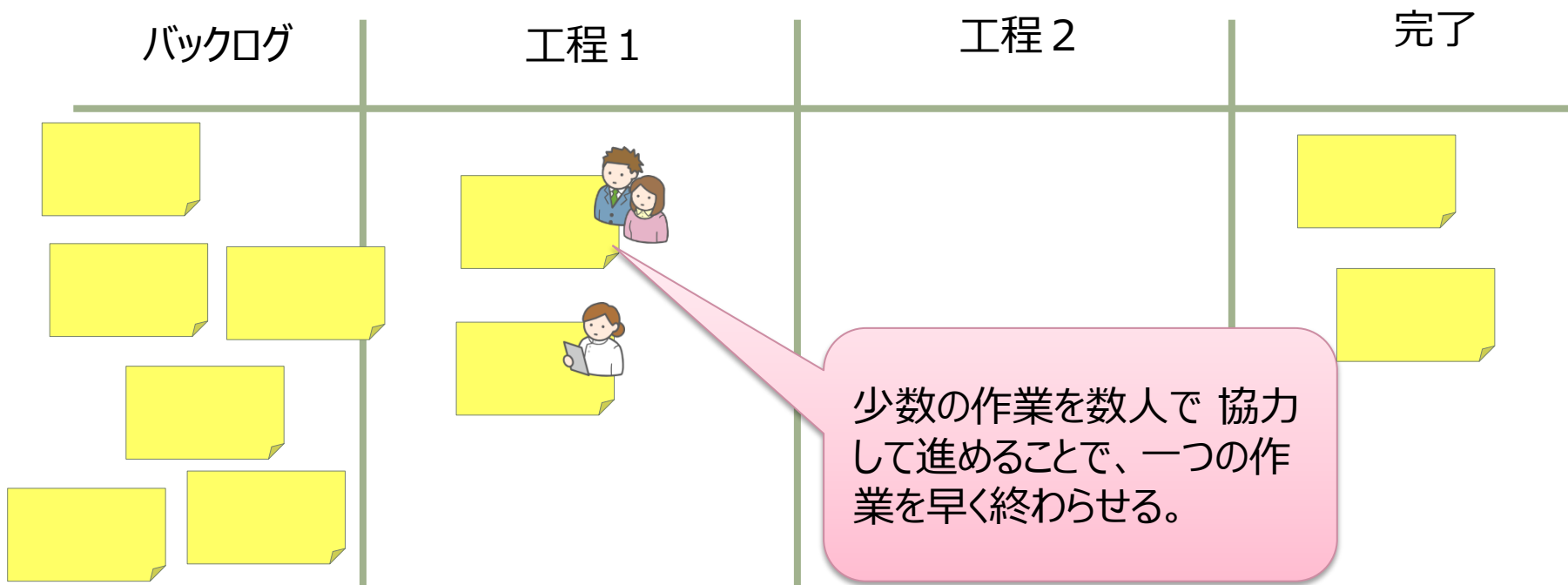
# 説明

# 演習 1 ではコラボレーションが起こらない



前工程が全て終わってから、次工程を開始する

# 演習 2 はコラボレーションが促進される



工程内の作業数（仕掛かり）に制限をかけ、  
少数の作業に集中して進める

# リトルの法則

# リトルの法則

店内の顧客数

単位時間あたりの  
来店人数

平均滞在時間

$$L = \lambda W$$

※安定な系であることが前提条件



# 行列の待ち時間を知る方法

行列の出来るラーメン屋で並んでいるときに、待ち時間が気になったとする。

行列内の顧客数

単位時間あたり  
新規に並んだ人数

行列にいる  
平均時間

$$L = \lambda W$$

1. 自分の前に並んでいる人数を数える (=  $L$ )
2. 自分の後ろに、1分間で新たに並んだ人の数を数える (=  $\lambda$ )
3.  $L \div \lambda$  が待ち時間 (=  $W$ ) !!



# 開発作業に当てはめると

仕掛け作業数

$L$

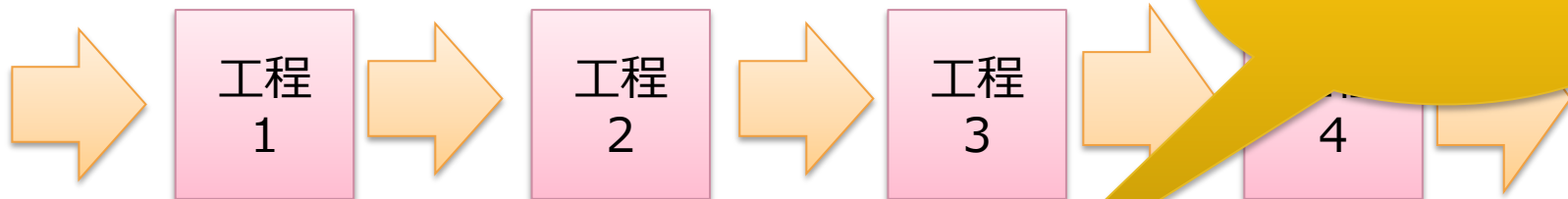
処理能力  
(ベロシティ)

$\lambda$

サイクルタイム

$W$

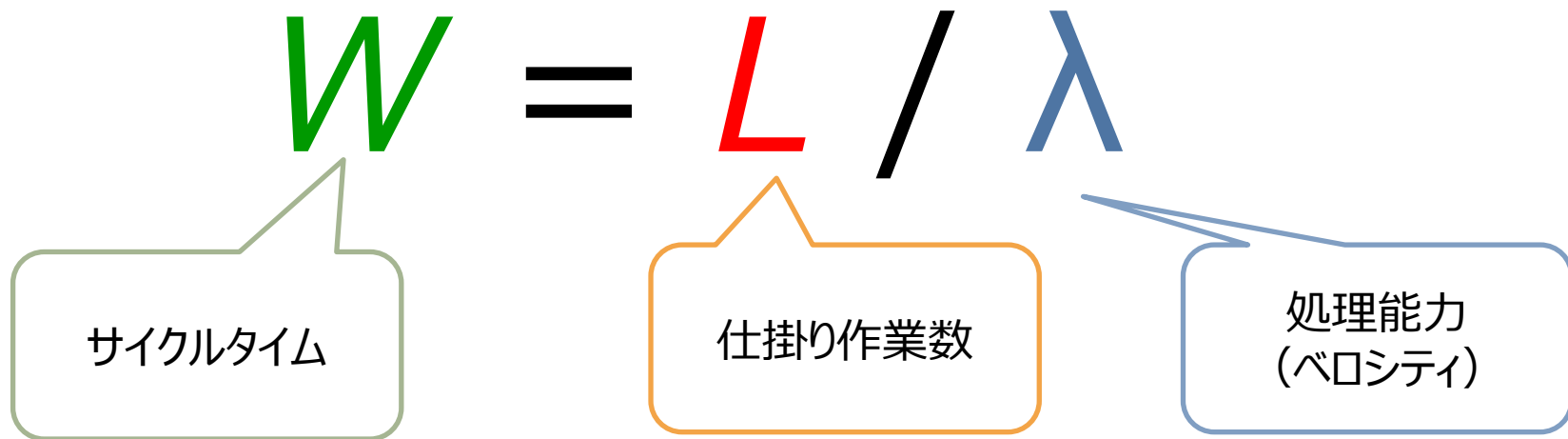
なるべく短くしたい！



$W$  (サイクルタイム)

# サイクルタイムを短縮するには

- リトルの法則の式を変形。



サイクルタイムを小さくするには、仕掛け作業数を小さくするか  
開発者の作業能力を上げるしかない。



仕掛かり作業数のことを

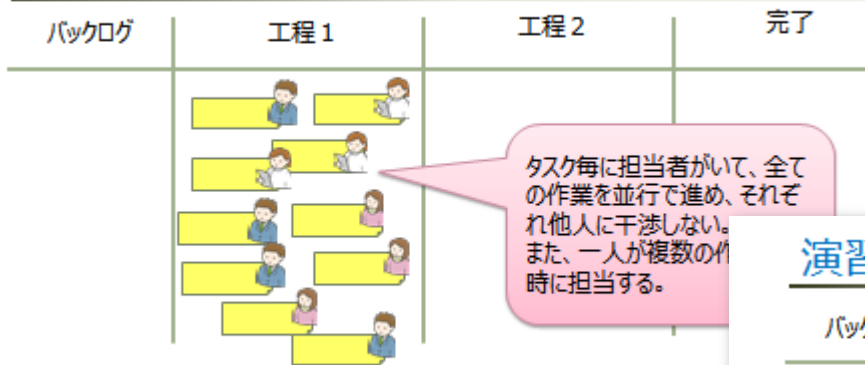
**WIP**

(Work In Progress)

という

# WIP制限

## 演習 1 はコラボレーションがない場合

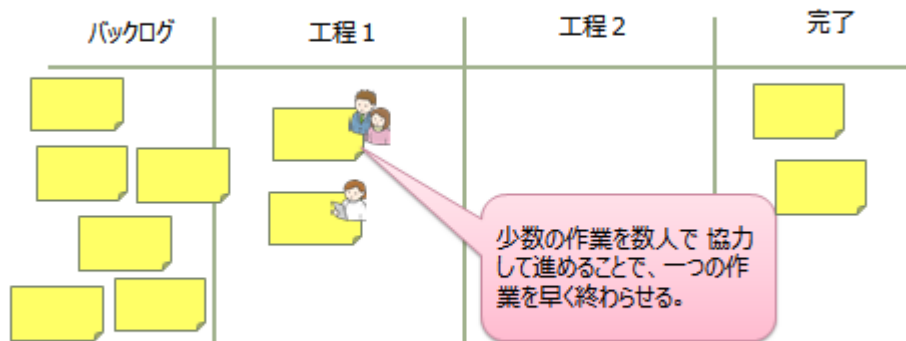


タスク毎に担当者がいて、全ての作業を並行で進め、それぞれ他人に干渉しない。また、一人が複数の作業時に担当する。

前工程が全て終わってから、次工程を開始する

演習 1 ではWIPを無制限にしていた

## 演習 2 はコラボレーションがあった場合



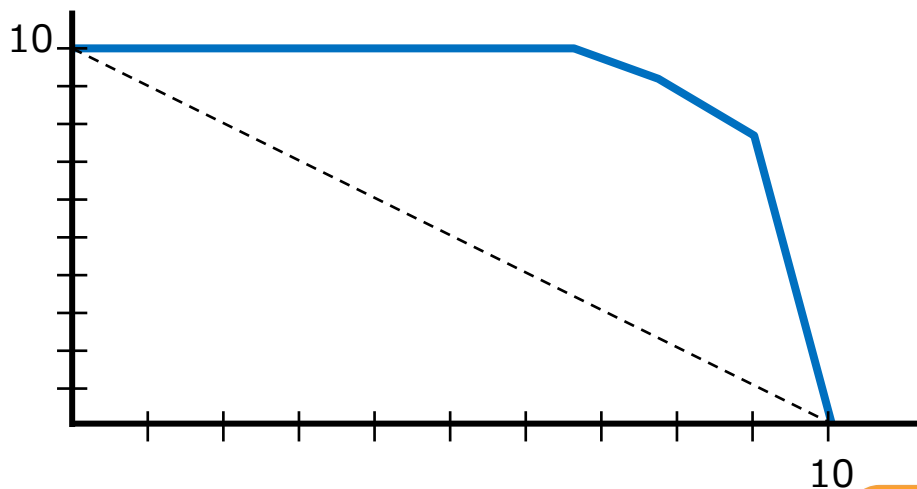
少数の作業を数人で協力して進めることで、一つの作業を早く終わらせる。

工程内の作業数（仕掛かり）に制限をかけ、少数の作業に集中して進める

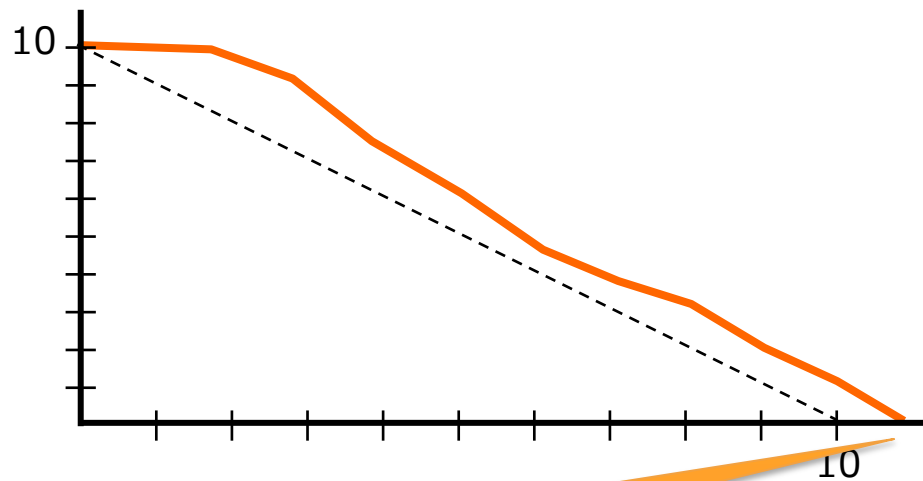
演習 2 ではWIPを“2”に制限していた

# 典型的なバーンダウンチャート

演習 1



演習 2



こっちの方が遅い!?

# キャッシュフロー

商品代金は  
100万円です



事業主

25日までに50万  
返済してください



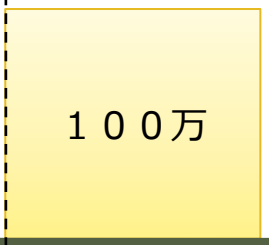
銀行

**返済期限**

月末(4週後)にまとめて  
100万支払います



買主



第1週

第2週

第3週

第4週

第5週

翌月第1週

毎週20万円ずつ  
支払います



買主

20万

40万

20万

60万

20万

80万

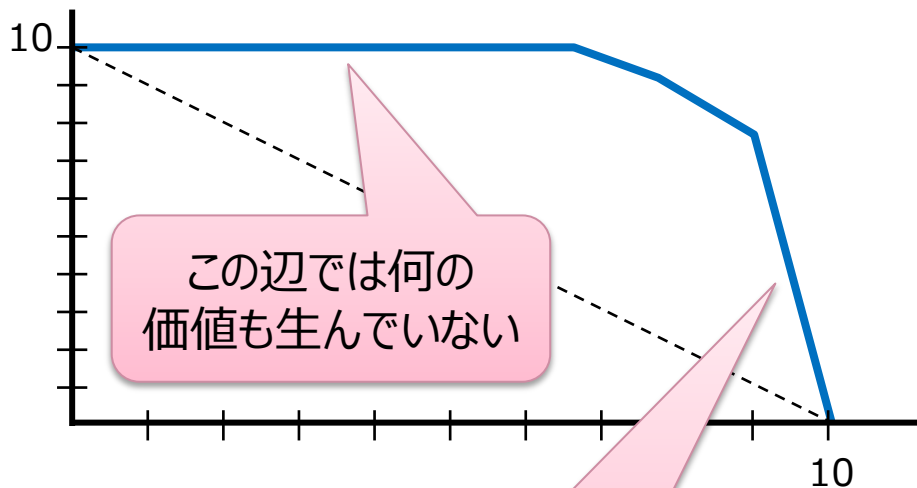
20万

100万

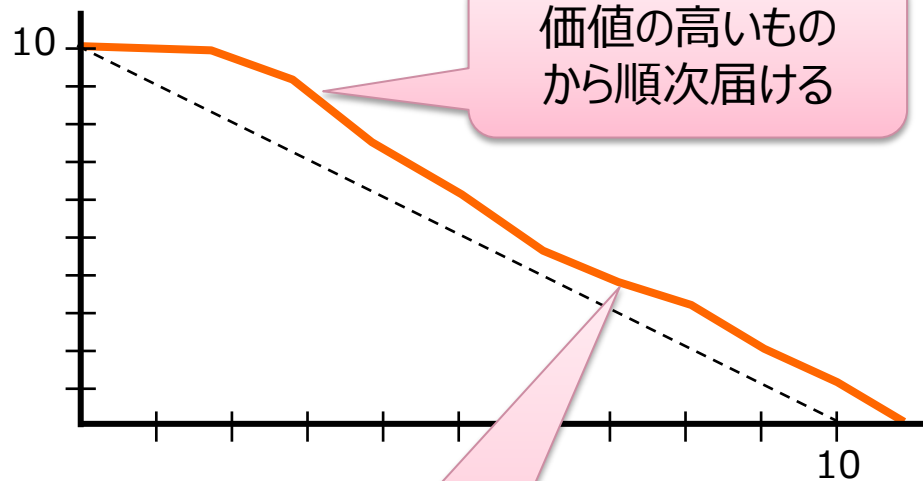
20万

# 価値のデリバリー

## 演習 1 ウォーターフォール的



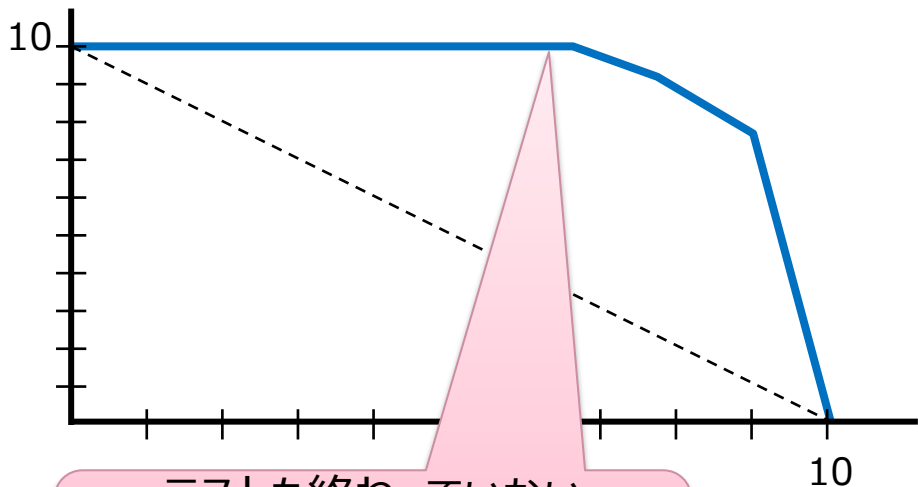
## 演習 2 アジャイル的



価値は常時  
継続的に届ける

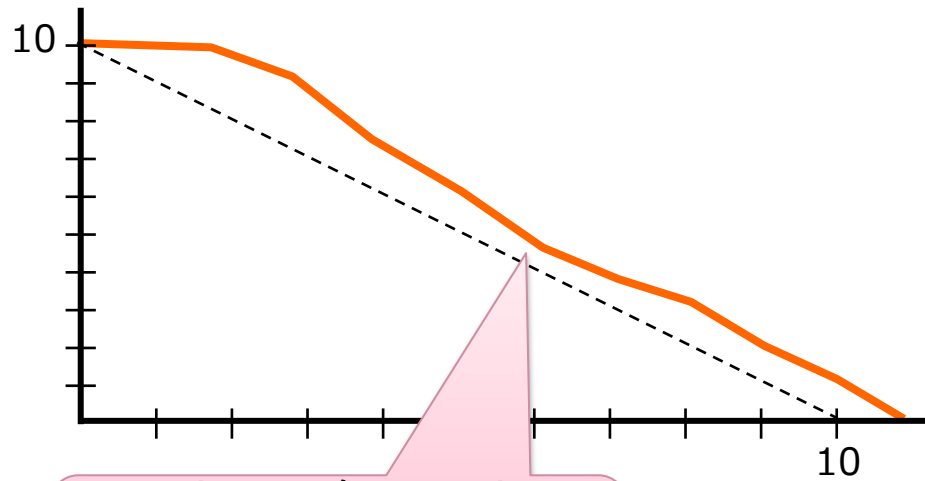
# 予測性

## 演習 1 ウォーターフォールの的



テストも終わっていない  
ストーリーが大量にあるため

## 演習 2 アジャイル的



ストーリー毎にテスト  
まで完了させているので

顧客にとってどちらが高い価値を生むのかを理解してもらうことが必要

# コラボレーション

---

- 今回は主にチーム内のコラボレーション（協働）について体験していただきました。
- が、コラボレーションの相手は、ステークホルダー、顧客、上司、セールス、製造事業所、QA、などなど多岐にわたります。

アジャイル開発は  
コラボレーションによって  
正しく真価を発揮する