

スクラムマスター 2年生の軌跡

～日中合同スクラム開発の現場からのレポート～

NECソリューションイノベータ

細川 雅貴

Orchestrating a brighter world

未来に向かい、人が生きる、豊かに生きるために欠かせないもの。
それは「安全」「安心」「効率」「公平」という価値が実現された社会です。

NECは、ネットワーク技術とコンピューティング技術をあわせ持つ
類のないインテグレーターとしてリーダーシップを発揮し、
卓越した技術とさまざまな知見やアイデアを融合することで、
世界の国々や地域の人々と協奏しながら、
明るく希望に満ちた暮らしと社会を実現し、未来につなげていきます。

自己紹介

■ 細川 雅貴(ほそかわ まさき)

■ 30歳の2児の父

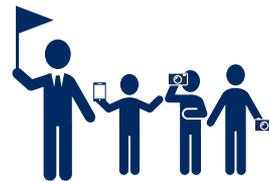
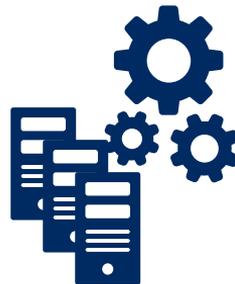
■ 職場は、広島県



■ 趣味は、子どもと一緒にポケモンGO

■ 以下のような仕事をしています

- 運用管理ソフトの開発(スクラムマスター)
- 深層学習を用いた人物認識
- 地域創生をキーワードとした新規事業発掘



目次

1. スクラム開発実践プロジェクトの紹介
2. スクラム開発の実践歴
3. スクラム開発を採用した理由
4. スクラム開発の実践の記録
 - 導入期
 - 成長期
 - 発展期
5. まとめ

本題に入る前に
質問です！

AgileJapanの参加は

1. 初めて
2. 2回目
3. 3回以上

今日聞いたセッションの数は

1. 1つ(今日初めて)
2. 2つ
3. 3つ以上

スクラム開発について

1. 知らない
2. 知っている
3. 実践した事がある

1.スクラム開発実践プロジェクトの紹介

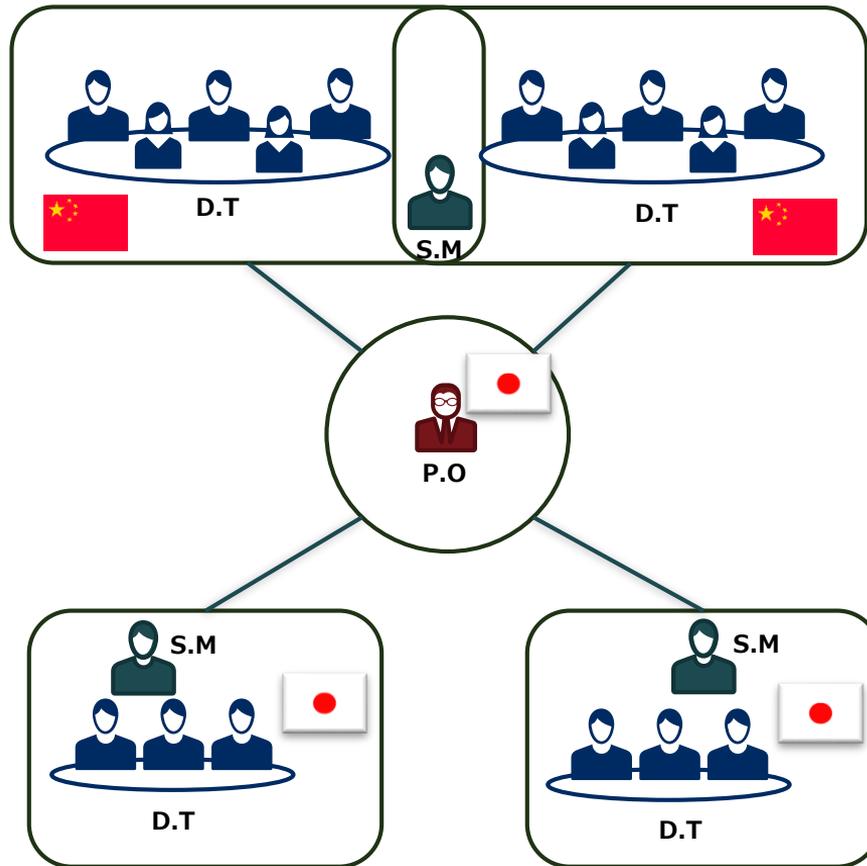
1.スクラム開発実践プロジェクトの紹介

プロジェクト概要

- 運用管理製品の新機能開発

体制

- 3つの開発拠点が連携して開発を進めている
 - ・ 日本(東京、広島)
 - ・ 中国
- 日本の開発チーム人数は、平均3人ほど
- 中国の開発チーム人数は、平均5人ほど



2.スクラム開発の実践歴

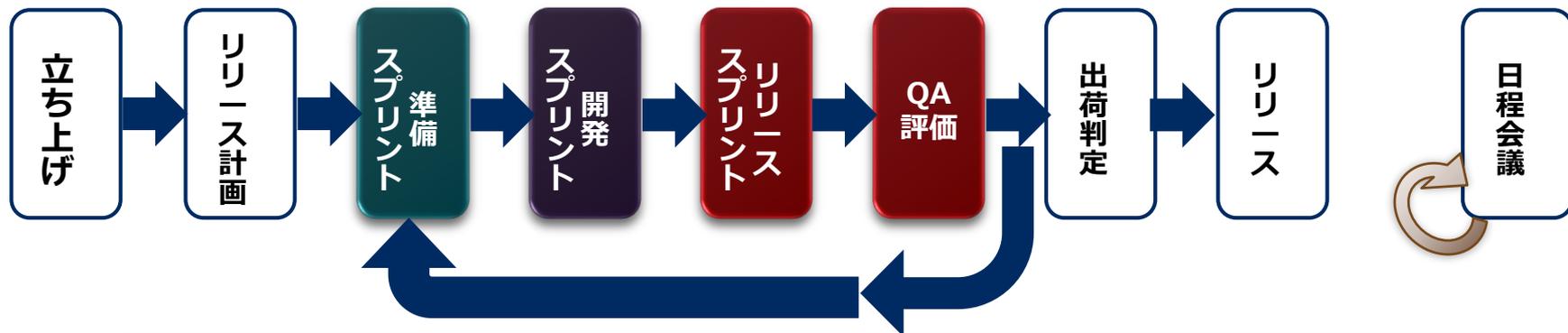
スプリント期間

2 週間

スプリント実施総数

37
スプリント/チーム

2.スクラム開発の実践歴 ～全体の流れ～



スプリントの基本アクティビティ



3.スクラム開発を採用した理由

3.スクラム開発を採用した理由

採用に至るまでの背景

- ウォーターフォール開発の終盤に問題が顕在化し、約1ヶ月の出荷遅延
 - ・開発終盤で仕様齟齬に気付けなかった。
 - ・仕様齟齬などの問題を早期発見する仕組みが弱かった。
- **開発終盤にならないと動くソフトウェアの評価が難しい**
- 大きな仕様齟齬が発生すると、スケジュールの取り戻しは非常に困難

採用した理由

- 自分達の開発の**問題点を素早く知る**仕組みが必要
- 他の企業で成功を収めている開発手法の取り込み

4.スクラム開発の実践の記録

スクラム開発 導入期



スクラム開発を**理解しないまま**、
開発チームを編成した。
とりあえず、スプリントやってみた！

仕様検討に時間がかかり
動くソフトウェアは生まれなかった

つまり、**成果ゼロ**

失敗した原因

スクラム開発を理解しない人が集まって
スクラム開発もどきをやったから

Let's 改善！

- スクラムマスター研修へGO
- リリース計画書の作成
 - スケジュール、体制、Doneの定義の決定
- プロダクトバックログの作成
- 開発インフラの整備

Redmine Scrum Pluginでスクラム開発



チャットツールZulipの導入



情報集約

スクラム開発が回りだした！





スクラム開発 成長期

開発チームの生産性(ベロシティ)に
ムラがあって**安定しない**

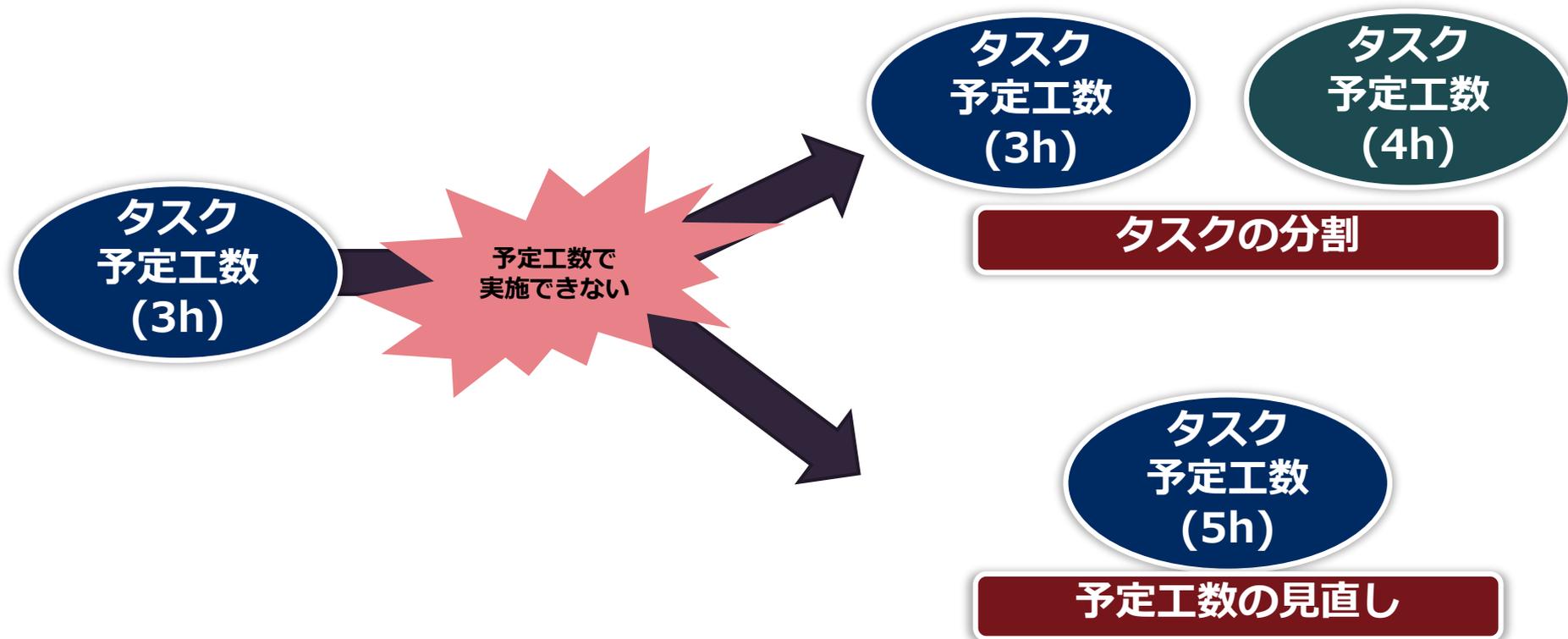
安定しない理由を調査した結果

- ・タスクの**予定工数の超過**
- ・スプリント**終盤で問題の顕在化**
- ・**アーキテクチャ設計**が場当たりの
- ・開発外作業が多い

セレモニーの日程調整、ストーリーの詳細の確認

Let's 改善！

タスクの予定工数の超過



スプリント終盤で問題の顕在化



	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日
プロジェクトの進捗状況を確認する															
タスクA															
タスクB															
タスクC															
タスクD															
タスクE															
タスクF															
タスクG															
タスクH															
タスクI															
タスクJ															
タスクK															
タスクL															
タスクM															
タスクN															
タスクO															
タスクP															
タスクQ															
タスクR															
タスクS															
タスクT															
タスクU															
タスクV															
タスクW															
タスクX															
タスクY															
タスクZ															

アラームの挙げ方の決定

問題の予兆を素早く検知

線表の共有

アーキテクチャ設計が場当たりの

アーキテクトの設置

ロードマップの共有

開発外作業が多い

- ・ セレモニーの日程調整

スクラムセレモニーの
スケジュール固定化

- ・ ストーリの詳細の確認

ストーリーの対象範囲と
対象外範囲を明確にする

おっ！

スクラム開発が安定してきた！



スクラム開発 発展期



複数のチームで

1つの目玉機能を開発しよう

目玉機能を早く完成させよう

想定していたより
開発がうまくいかない。。。。

Why!? Scrum people!!

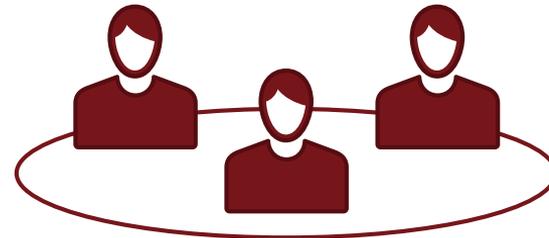
うまくいかない要因

- ・ **チームを超えた**情報共有が不十分
- ・ **ストーリー間の依存関係の把握**がより必要になった
- ・ 開発メンバーの入れ替わり
- ・ **2スプリント先の未来**しか見えない

チームを超えた情報共有が不十分



仕様メモ

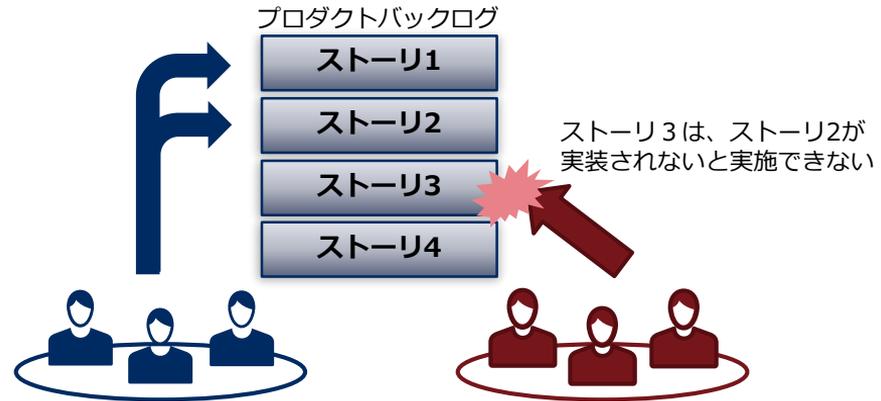


仕様メモに記載すべき内容とは？

F2Fのコミュニケーションの情報量 >>> 仕様メモの情報量

同一チーム内の情報共有は問題ないが
複数チーム間の情報共有方法が確立できていない

ストーリー間の依存関係の把握がより必要になった

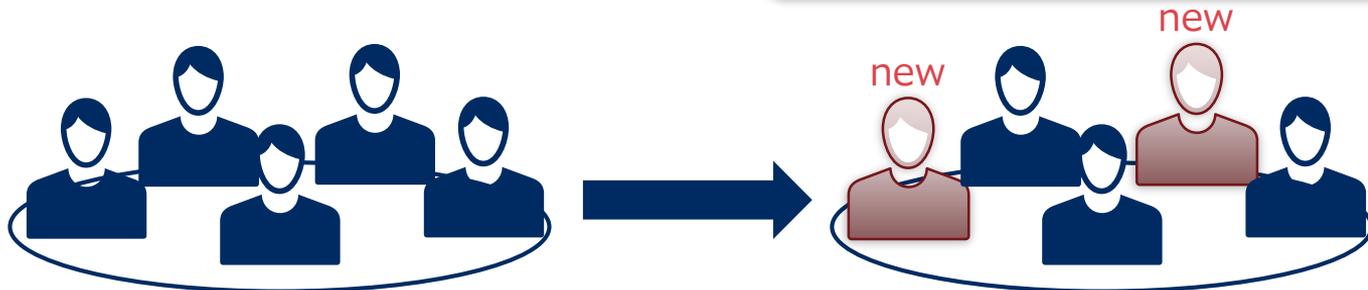


ストーリーの分割による依存関係の発生を把握する必要がある
コードコンフリクトの懸念も考慮する必要がある

開発メンバーの入れ替わり

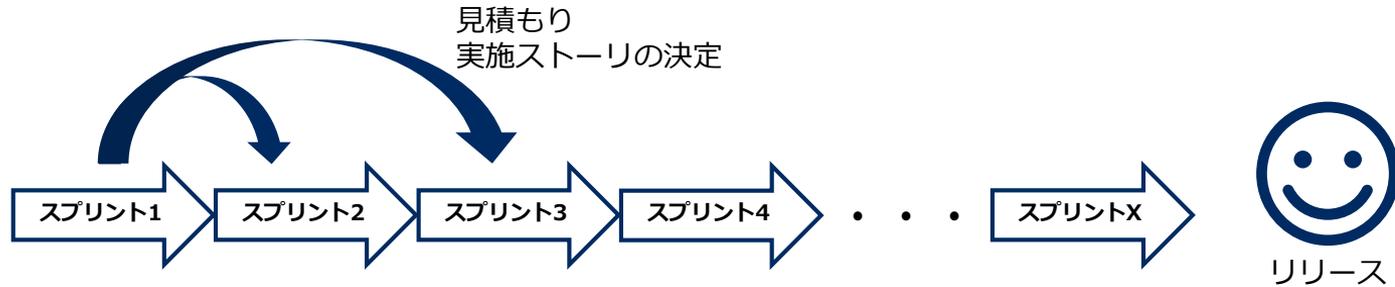
中国の場合、結婚、出産を機会に転職する場合がある

開発メンバーが変わるリスクをどう最小限にするか？



新しい人が入る事でチームの自律性が一時的に損なわれる
技術的にも新人の場合は、生産性が安定しない可能性がある

2スプリント先の未来しか見えない



生産性(ベロシティ)の安定後、コア機能のストーリーが
全て消化可能か検証する必要がある

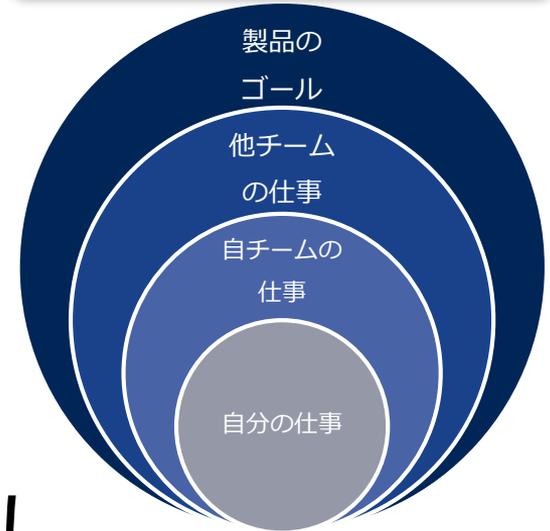
誰がどのタイミングで行うか？

Let's 改善！
だけど、まだ模索中

複数チーム構成の場合、
自分のチームだけではなく
他のチームへ意識を向ける

また、自分達のゴールを意識し
常に**達成可能かどうか**を見定める

意識する範囲の拡大



5.まとめ

開発手法の**変化**が

開発メンバー一人一人の**変化**を引き起こす

開発メンバー一人一人が**変化**すると

開発、製品が**変化**する

変化を受け入れて**進化**しよう

 **Orchestrating** a brighter world

NEC