\Orchestrating a brighter world NEC

オフショア連携とスクラム開発の両立

～自分ごとで考えるチームを作る～

NECソリューションイノベータ株式会社　プラットフォーム事業本部
安田 康士

\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。

目次

今日、聞いて欲しい人

こんな事、ありませんか？

開発チームが
各地に分散

オフショア
異文化交流

保守的な
社内ルール

ピラミッド型
指示待ち体質

取り組みの背景

ミッションクリティカル製品の開発

ミッションクリティカル
(英：mission critical, mission-critical)

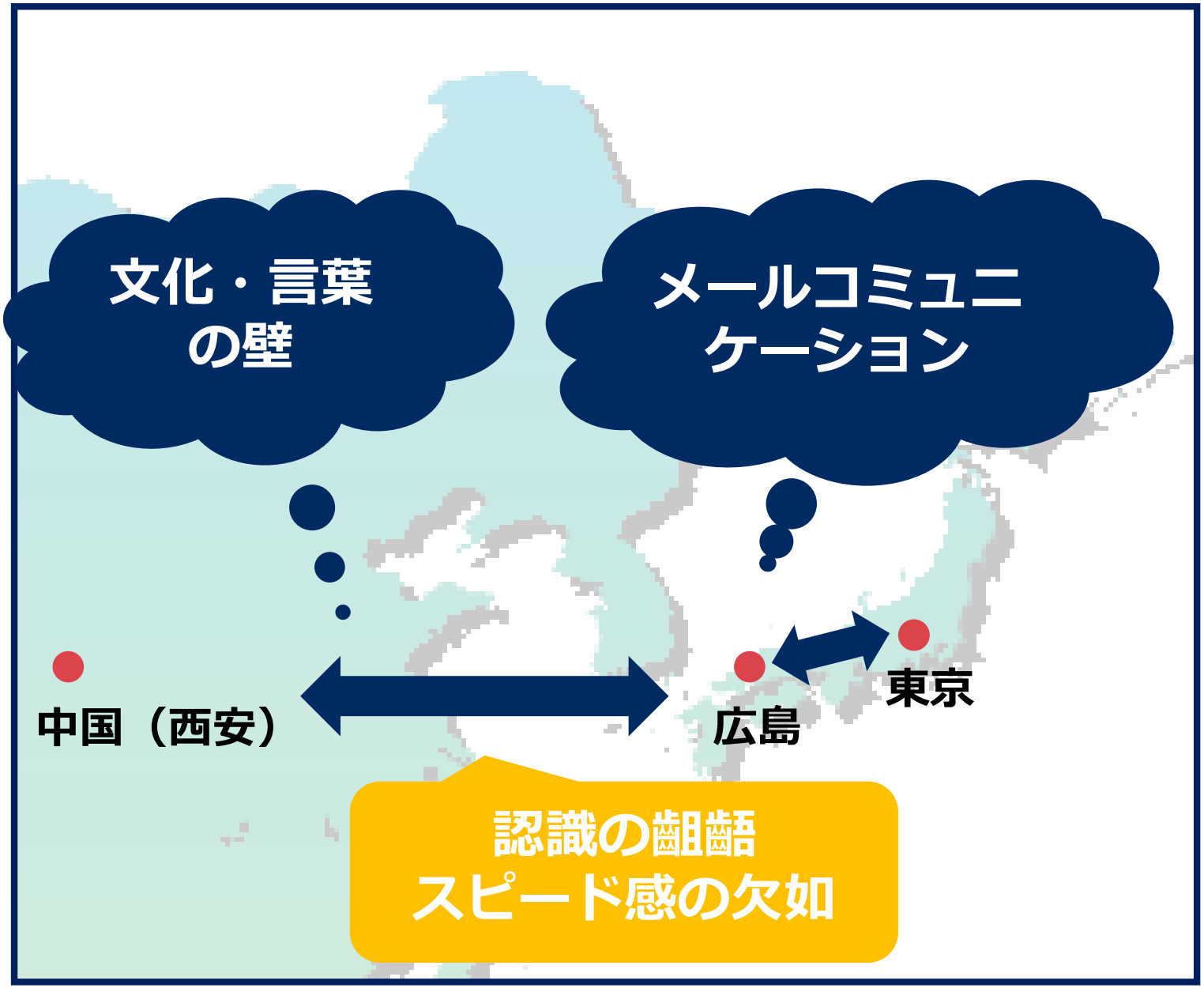
任務や業務の遂行に必要不可欠な要素（機器、プロセス、手順、ソフトウェアなど）のこと。障害の発生よる中断や停止が発生した場合に社会的影響が大きい、交通機関や金融機関などの基幹システムは一般にミッションクリティカルであり、停止しないことが求められる。

https://ja.wikipedia.org/wiki/ミッションクリティカル

品質に求められる水準が高い開発

私たちのチームの開発プロセスの変遷

分散開発



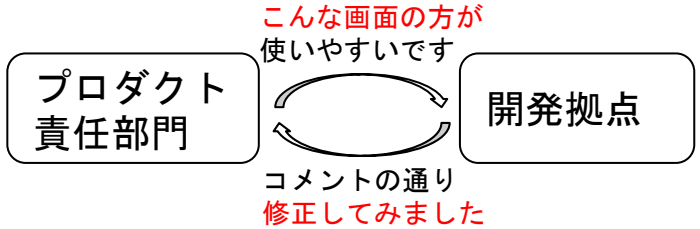
過去の失敗事例・課題

ウォーターフォール開発

製品に**利用者視点の課題**が蓄積しやすい

- **開発終盤に仕様の認識齟齬が発覚**し、スケジュールの大幅な遅れが発生
- 要件定義後に変更を受け入れないため、**利用者視点に立った改善を途中から盛り込むことができない**
- イテレーティブな機能改善の**サイクルが長く**、保守コストが増大

プロトタイプ + ウォーターフォール開発



- **意図や全体像をつかめず**パッチ的なプロトタイプ作成
- システム評価の品質指標にも影響し、スケジュールの大幅な遅れ

ウォーターフォール開発やプロトタイプ開発が悪いわけではない

- **ビジョンが共有できていない**
- **製品の価値提供方針に開発スタイルが適合していない**

失敗事例分析

指示者側の問題 - 指示・要求がハイコンテキストである

例) ユーザー管理機能をLDAPと連携できるようにしてください

- 「誰が」新しい機能利用するのか？ (利用者像)
- 「どのように」あたらしい機能がお客様に利用されるのか？ (ユーザーストーリー)
- 「将来」この機能が何に生かされるのか？(ロードマップ)

オフショアのメンバーが隠れたコンテキストを読み取るのは容易ではない

被指示者側の問題 - 開発終盤に機能実装不足や品質問題が発生する

- 要求通りの動作だと考えていた
- 周辺機能の動作について把握していないので考慮していなかった

自分達自身が問題を認識できていなければ対処できない

- **スキルの違い:**
  - **ソフトウェア開発を志して入社したメンバーが多いため、システム運用の知識に疎い**
  - **人の流動性が高く、チームにスキルや知識が定着しない**
- **国・文化の違い:**
  - **同じ言葉に対する認識の違いがある**
  - **質問をしたがらない(メンツを重視する)**

改善したいこと

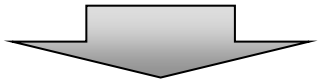
改善したいこと

課題1:
製品の現状と未来のビジョンを共有する必要がある

- 距離や文化、言語によるコミュニケーションの障壁を下げる
- 遠隔拠点を含むチームの合意形成スピードをアップ
- タスクの背景にある情報を集約する

課題2:
強化プロセスにあった開発スタイルを取り入れる必要がある

- イテレーティブな、柔軟さを持った開発スタイルにしたい
- 継続的に変化(改善)を取り入れ続けたい



反復 (イテレーション) と呼ばれる短い期間単位で開発する
アジャイル開発手法の一つであり、現状を把握するフレームワーク
であるスクラム開発を適用する

スクラムの概要

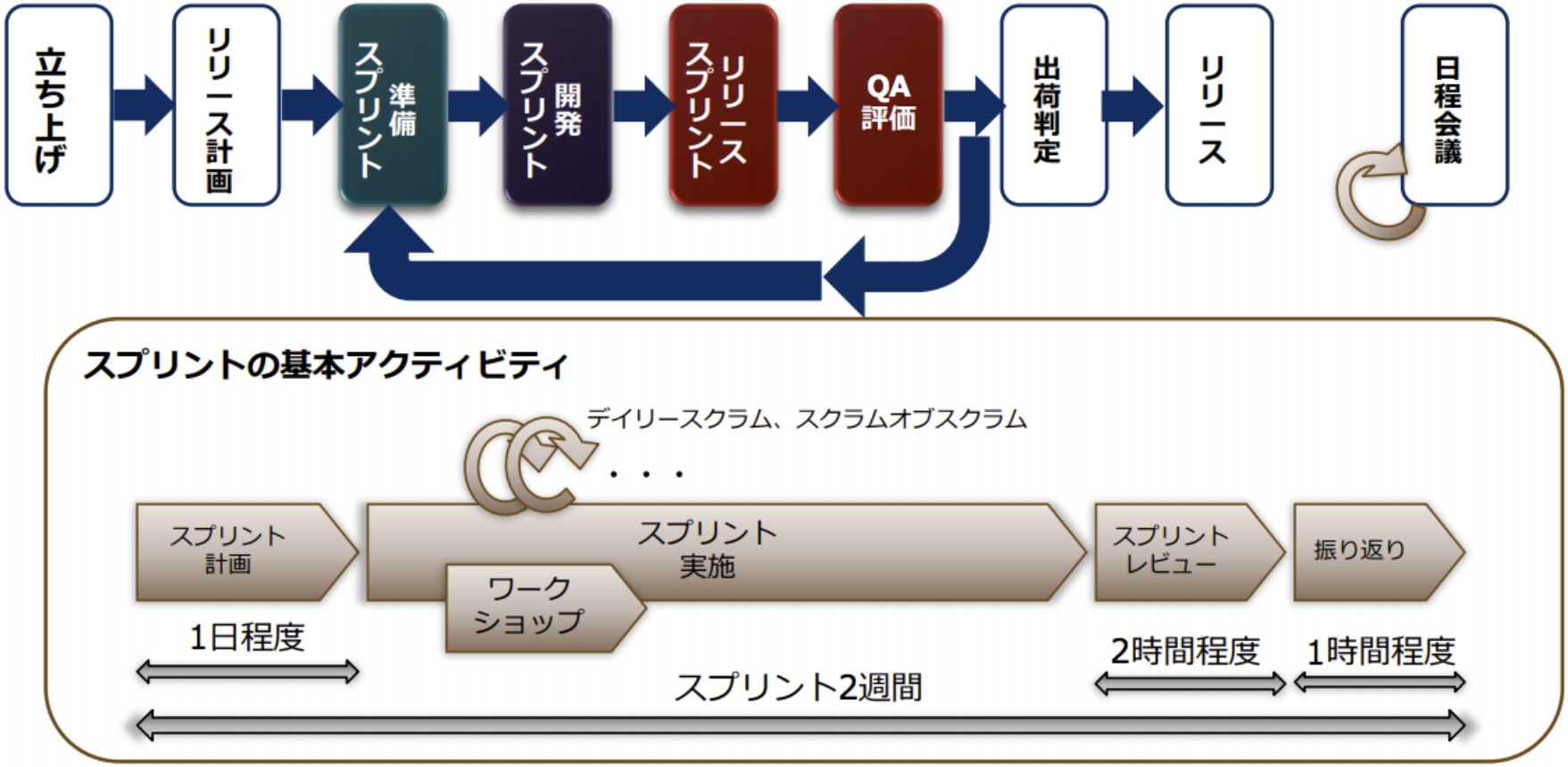
スクラムとは、経験主義にもとづき反復的かつ漸進的にプロセスを進め、現状を可視化するフレームワーク、方法論。

もっとも採用されているアジャイルソフトウェア開発手法であり、この方法論は、チームが自発的に組織だって行動することを可能にする。

経験的プロセス制御の実現は、透明性(Transparency)、検査(Inspect)適応(Adapt) の 3本柱に支えられる。

3 つの柱	
透明性	鮮度の高い情報(状況、問題点)を常に明らかにし、スラムチームの現状や問題点を可視化すること。
検査	スクラムチームの現状や問題点がないかを確認し、共通理解ができること。
適応	スクラムチームに問題があった場合に、改善策を考えて対処すること。

スクラム実践の流れ



改善策の内容

課題1への取り組み

製品の現状と未来のビジョンを共有する必要がある

- 距離や文化、言語によるコミュニケーションの障壁を下げる
- 遠隔拠点を含むチームの合意形成スピードをアップ
- タスクの背景にある情報を集約する

宣言

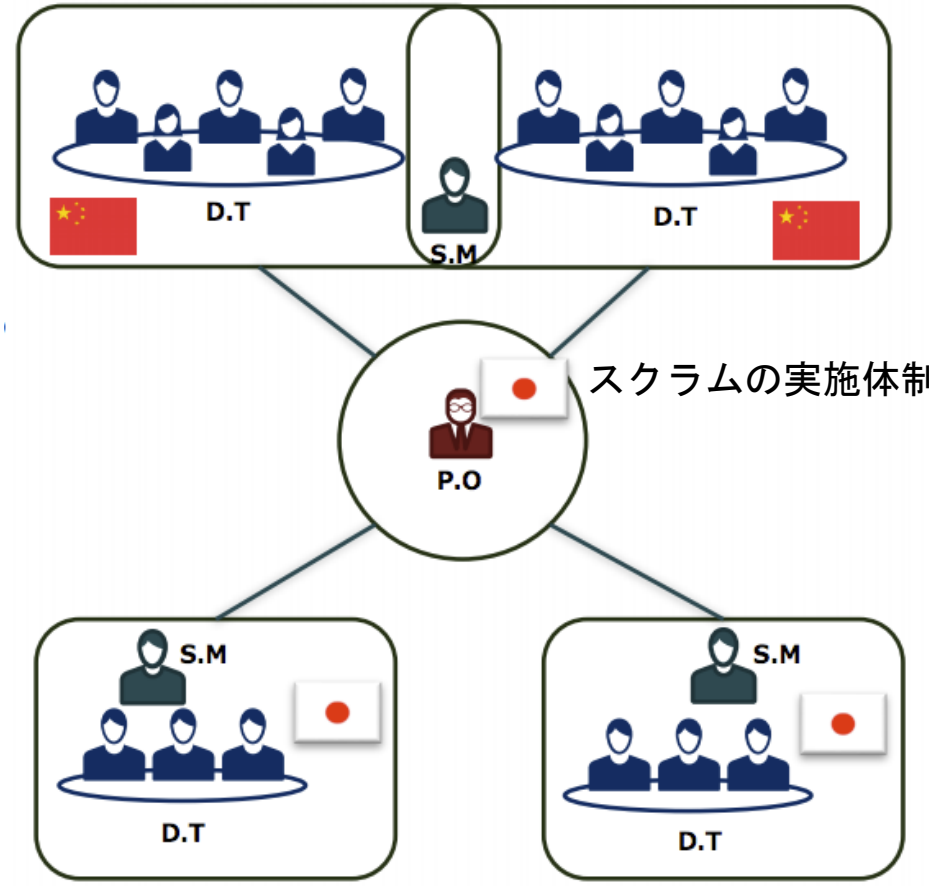
コミュニケーションの障壁を下げる

スクラムの価値基準

スクラムチームが、確約（commitment）・勇気（courage）・集中（focus）・寛容さ（openness）・尊敬（respect）の価値基準を取り入れ、それらを実践するとき、スクラムの柱（透明性・検査・適応）は現実のものとなり、あらゆる人に対する信頼が築かれる。

寛容さ、尊敬を実践する土壌づくり

- スクラム開発を選択した目的・理由をステークホルダー全員で共有
- 各メンバーがフラットな体制であることを宣言
- 意見・変化・失敗を許容・歓迎することを宣言



スクラムの実施体制図

コミュニケーションの改善（Before）

コミュニケーションの障壁を下げる
合意形成スピードをアップ

従来のコミュニケーション

Eメール

- ○ オフショアメンバーも日本語を読むことができるメンバーは多いため実担当者と直接議論を交わすことができる
- ✖ リアルタイムなやり取りには時間的オーバーヘッドが大きい
- ✖ 体裁を整えてメッセージを送るため温度感が伝わりにくい

情報の発信・共有には向くが、議論するには効率が悪い

電話/Skype/リモート会議

- ○ リアルタイムにやり取りできる
- ✖ 日本語で会話できるオフショアメンバーがリーダー層に集中する
- ✖ 実担当者への展開は伝言ゲームになりがち

議論向きだが、過程が全メンバーに共有しきれない

コミュニケーションの改善（After）

コミュニケーションの障壁を下げる
合意形成スピードをアップ
タスクの背景にある情報を集約する

チャットツールでのコミュニケーションへの統一

- ○ リアルタイムなやり取りにも同報にも利用できる
  (記録的な同報はメールも併用)
- ○ emoji reaction (ありがと 3 👍 2)を利用してコミュニケーションの敷居を低くする
- ○ 複数のツールからの通知も集約
- ○ トピックの区分が「全員が同じ」になり、認識を合わせやすい
- ○ 「あの情報」「この情報」がリンクで共有できる点も大きなメリット

課題2への取り組み

強化プロセスにあった開発スタイルを取り入れる必要がある

- イテレーティブな、柔軟さを持った開発スタイルにしたい(→ スクラム開発を適用したい)
- 継続的に変化(改善)を取り入れ続けたい

試験施行からのフィードバック

スクラム開発を適用したい

問題は必ず発生するものである。振り返りを通して継続的な改善が進むことを指標として計測していく。

- ●各メンバーが自分事として考える
- ●メンバーが抱えている課題点をすべて洗い出す
- ●課題に対する適切な解決策を導き出す
- ●解決策を実行する

スクラム開発導入期の取り組み (1)

スクラム開発を適用したい

問題	課題	施策
スクラム開発に対する理解の不足 オフショアメンバー混合チーム１スプリントだけ試験実施 ⇩ 確認に次ぐ確認で成果物はゼロ。プロセスをなぞるだけのまねごとでは成果は得られない。	スクラム開発ってそもそも何？ • メンバーの役割がわからない • スクラム開発のルールがわからない • 短期間(2週間のスプリント期間)の間に実施するべきタスクが作れない	各拠点キーマン(オフショア含む)の社外講習受講および認定資格取得  勉強会、外部PJのスクラムチームにヒアリング 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

スクラム開発導入期の取り組み (2)

スクラム開発を適用したい

問題	課題	施策
仕様検討に時間がかかる	P.O.が確認できずボトルネックになる	各拠点にエリアP.O.を設置
		スプリント毎に振り返り(KPT)実施 小さな改善を自ら決めて実践
	開発外作業が多い **スクラム開発の恩恵(透明性)によって見えるようになった課題**	全拠点の状態を可視化できるようにツールを整備 → 課題1 で取り組み

スクラム開発成長期の取り組み

スクラム開発を適用したい

問題	課題	施策
開発チームの生産性にムラがあって安定しない	機能追加に耐えられないアーキテクチャで設計してしまう	技術責任者を設置する
	ストーリの粒度が詳細化され過ぎ、全体像が見え難い	機能・開発ロードマップを作成して共有する
	タスクの予定工数を超過してしまう	タスクを小さく正確に分割する 予定工数をデイリーに見直す
	開発メンバの入れ替わりの発生 (主にオフショアにおける離職問題)	仕様書メモに結果だけでなく経過も残す → 「なぜ」の確認削減

受け身の立場からの脱出

継続的に改善を取り入れたい

オフショアのメンバーはイノベーティブな発想を持っていても立場を線引きしてフィードバックをためらうことが少なくない。新入社員や経験の浅いメンバも同様の傾向がある。

自分事として捉え、改善したいと提案できる環境とは？

問題や疑問を感じとる**場や機会**を増やす

- チーム内の代表役を順番にアサイン
  - スクラムセレモニーの司会進行
  - チーム全員のタスク、進捗
  - ステークホルダとの日程調整
  - スプリントレビューでプロダクトオーナーにデモ／説明

→「作業の障害になる点は排除したい」

- 問いかけ
  - 「どうすれば良いと思う？」

→「私の意見は・・・」

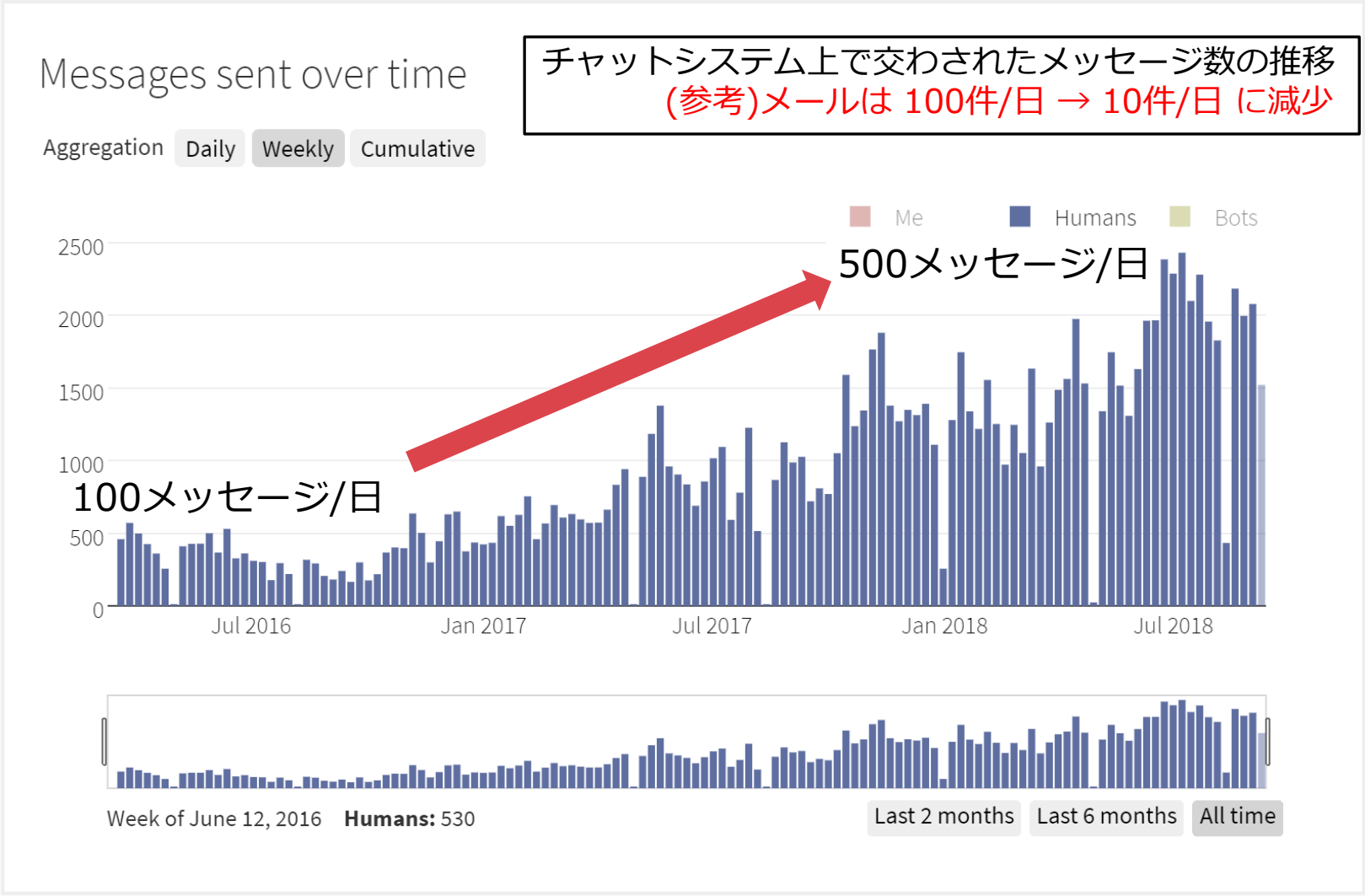
あえて**小さな失敗**を経験してもらう

→「なぜ先にノウハウにたどり着けなかったのだろう？」

改善による変化や効果

課題1(ビジョンの共有)への効果

コミュニケーション方法の変化(メール → チャット)によりコミュニケーションの頻度が大幅に上昇。細かな点の確認がためらいなく行われるように変化。

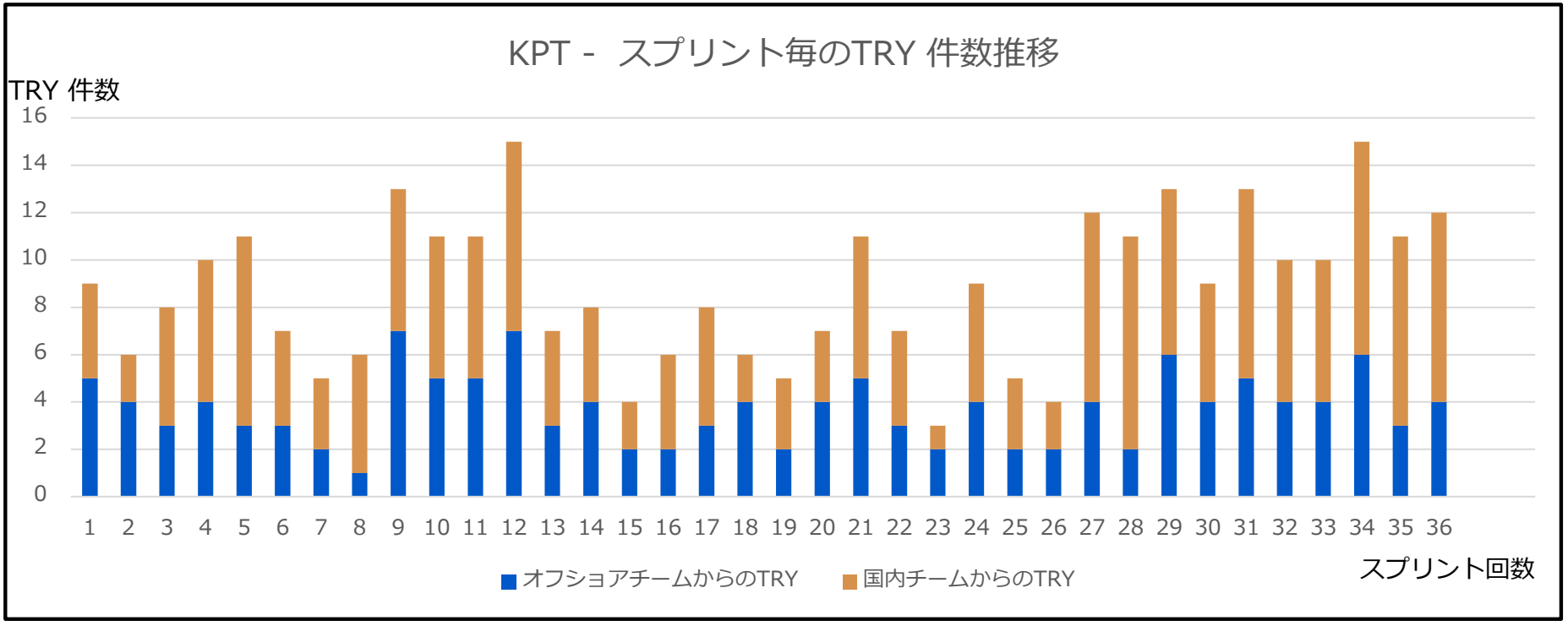


課題2(継続的な変化の許容)への効果

ウォーターフォール開発時代：振り返りは開発終了時に1度(1年に1回)

スクラム開発導入後：振り返りをスプリント毎に実施(2週間に1回)

- 忘れないうちに振り替えることで、小さな課題も改善
- オフショアメンバーも改善に積極的



メンバーの声

メンバの意識の向上

- スプリント単位の予定を立てるためリズムが生まれ、**先が見通し易くなった**
- 問題を改善し、**常に向上していく意識**がより強くなった
- **個人で問題を抱え込まずチームで解決**するようになった
- 視野が広がり、**一人一人が全体を意識して自発的に行動**するようになった

離れた拠点間のコミュニケーション活性化

- **コミュニケーションが素早く、より活発に**なった
- 質問をすることに気後れしなくなった

問題の可視化

- チームが抱えている問題を浮き彫りにしていくことができた
- 早く現状（問題）に気が付く事ができ、対策の検討実施を行う事ができた
- 開発に関するプロセスの明文化が進み、チームの効率が向上した

考察

スクラム開発の導入で、効果・期待できること

- チームが自立します
- リズムが生まれます
- 曖昧さが排除されます
- やるべき事に集中できます
- 問題が明らかになります
- 改善が癖になります

まとめ

まとめ

オフショアと共にスクラム開発に取り組み、

オフショアと共に改善する組織の知見を得た

- 開発者としてお互い対等だと実感をもつ
- 失敗を許容する (場合によってはあえて失敗させる)
- チャットツールなどを活用してコミュニケーションの敷居を下げる

ウォーターフォール文化（ピラミッド体制、指示待ちの姿勢）からの脱却を実践した

- 現状維持が楽 → 改善したほうが楽 になる体験を繰り返す
- 個人→ チーム → 他拠点 → 製品としてのミッション の順に
  大きな枠組みに目を向ける機会を繰り返し経験する

スクラムの価値基準(確約、勇気、集中、公開、尊敬)はオフショア開発の関係強化にも有効。

\Orchestrating a brighter world NEC

見えない未来は、
こころで見つける。

あなたの安心のために、あなたの夢見るチカラのために、
あなたの暮らしの土台になりたい。
私たちはまだ見ぬ課題もこころで発見し、
ICTをとおして笑顔が生まれる世の中を
創造していきます。

NECソリューションイノベータ


Orchestrating a brighter world
NEC