



NW自動化開発 × アジャイル/スクラム開発 ～理想と現実の狭間で～

伊藤忠テクノソリューションズ株式会社

渡部友也

- 自己紹介・課の紹介
- スクラム開発とは
- NW自動化チームがスクラムを採用した理由
- NW自動化とスクラム開発の親和性
- NW自動化チームのスクラム開発のリアル
- よくなったこと
- 悩んでいること
- 大事なこと
- 議論

- 渡部友也(わたなべともや)

- 東京都足立区出身
- 伊藤忠テクノソリューションズ所属(通称CTC)
- 2010年より現職で通信キャリア様を担当するエンジニア
- 高専時代からずっとITなので、気づいたら20年技術をやってます
- 趣味 競馬・酒・野球・麻雀
 - 競馬の回収率は80%しかありません。誰か助けて下さい。
 - コロナが5類になって初めてのJANOGなのでたくさんの人と飲めるの楽しみです
 - 野球はカープファンです。新井監督からチーム運営を学んでいます
 - 麻雀はコロナで雀荘に行くのが減りましたが、また再開したいです



- JANOGとの関わり

- JANOG48 IoT-GWプロビシステムの裏側
- JANOG49/50 参加のみ
- JANOG51 Booth
- JANOG52 Booth/本発表

- NW/インフラに付随する自動化チーム

- 自動化オーケストレータWeb(OpS)の設計・開発・維持管理
- 自動化エンジン(NSO, Terraform, Ansible等)のライセンス販売+操作用モジュールの設計・開発
- NWの検証自動化(Robot Framework)、出荷検査自動化開発および関連OSSの開発

Mission

Config EngineのAPIを統括する
Webオーケストレータ等の開発

NSOやAnsible/Robot Framework等の
モデル/PlaybookやAPI開発

Config Engineとの実機結合開発
NW検証の自動化(Robot Framework等)

Product



Value.2

要件整理支援および業務改善支援
OpS開発および工事業務のDX支援

コンフィグエンジンの技術サポート提供
自動化関連開発手法やナレッジの提供

マルチベンダーでの自動化
リグレッション試験の高度化

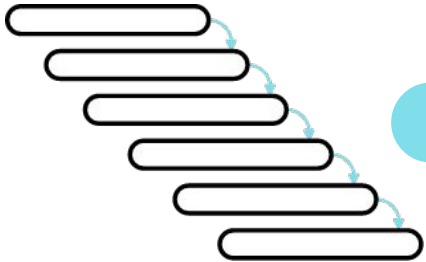
スクラム開発とは



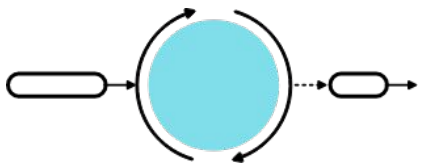
アジャイル？スクラム？

- アジャイル型開発とは開発における全体的なコンセプトであり概念である
- スクラム開発はアジャイルソフトウェア開発のフレームワークの一つ

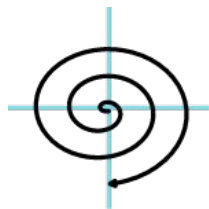
非アジャイル型開発



ウォーターフォール

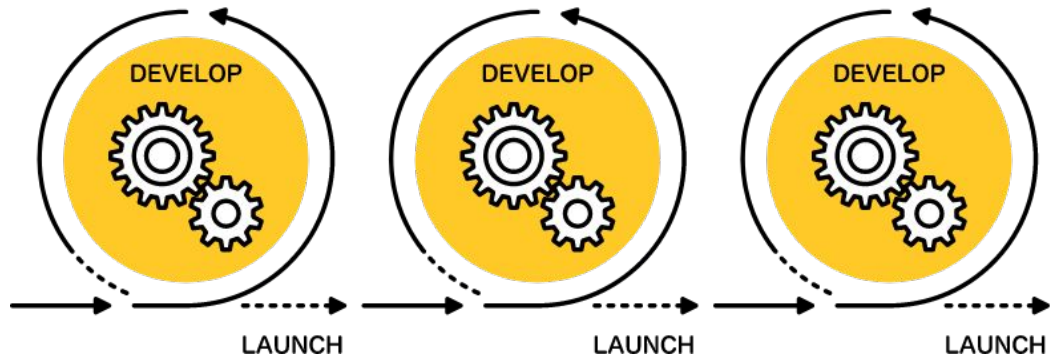


プロトタイプ



スパイラル

アジャイル型開発



スクラム

XP

FDD

リーン開発



アジャイルソフトウェア開発宣言



アジャイルソフトウェア開発宣言

私たちは、ソフトウェア開発の実践
あるいは実践を手助けをする活動を通じて、
よりよい開発方法を見つけだそうとしている。
この活動を通して、私たちは以下の価値に至った。

プロセスやツールよりも個人と対話を、
包括的なドキュメントよりも動くソフトウェアを、
契約交渉よりも顧客との協調を、
計画に従うことよりも変化への対応を、

価値とする。すなわち、**左記**のことがらに価値があることを
認めながらも、私たちは**右記**のことがらにより価値をおく。

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

© 2001, 上記の著者たち

この宣言は、この注意書きも含めた形で全文を含めることを条件に自由にコピーしてよい。

<http://agilemanifesto.org/iso/ja/manifesto.html>

より具体的にいうと？

プロセスやツールよりも個人と対話

レビュー指摘更新しました。ご確認よろしく
お願いします。

slack

XXXの件承認をお願いします。

XXに資料を配置しました。ご確認
宜しくお願
いします。

Jira

ツールを利用して
丁寧にコミュニケーション
している感じはあるけど、
内容は全く
会話できてないな...

包括なDocよりも動くソフトウェア

設計書や試験表
レビュー指摘表みたいな
ドキュメントはたくさん
あるけれど
実際に触れる人がいない...

契約交渉よりも顧客と協調

前提条件

免責事項

リスク

コロナによる影響

リスク・前提の話ばかりで、全然
前に進まないし
お客さんからすると
柔軟な対応とはいえない...

計画に従うことよりも変化に対応

再見積が必要！

聞いててた話と違う！

リスクー！

その手法は
再検討が必要！

計画の再検討！

計 画 なんて 変わるんだから
前に話してた話をたどるより
今からみんなで調べて
前に進めばいいのでは...

スクラムの主要な用語とイメージ



ビジョン
ネットワークAを自動化することでナレッジを貯める。ネットワークAで貯めたナレッジは他にも広げていく

プロダクトオーナー

プロダクトバックログ
まずはネットワークAで必要な技術習得・設計・開発を優先する！ネットワークBの検討はあとまわしにする！

スクラムマスター

スプリントバックログ
このスプリントでは技術習得を先にしますね設計までは到達しないかも

スプリントバックログ
PythonのLibraryの調査と現行ネットワークの装置の設定内容をちゃんと把握する必要がありますね

開発チーム

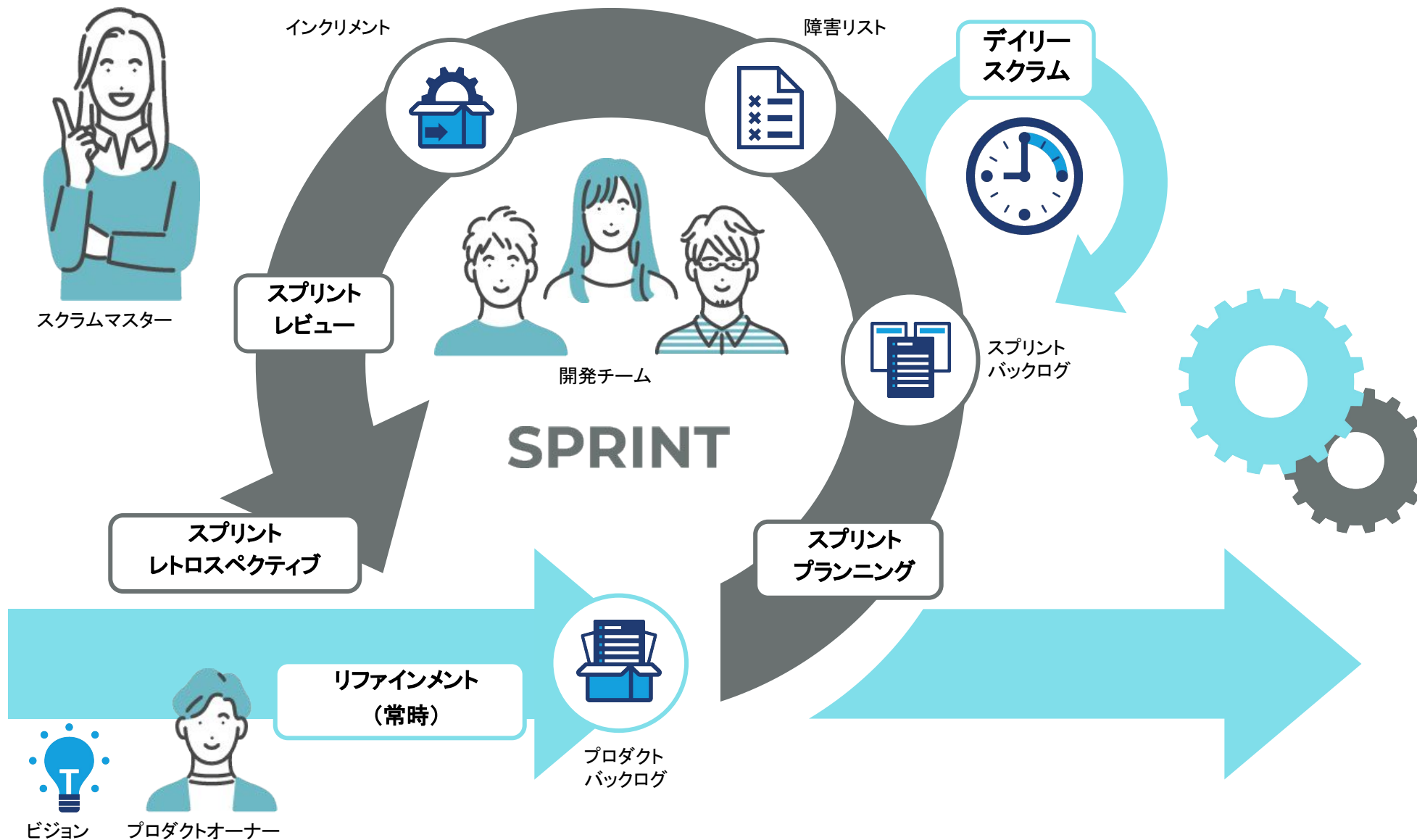
障害リスト

障害リスト改善
開発プロセスを効率良くするために開発環境を改善しよう

障害リスト
開発機のスペックが低いから、検証などの業務に支障をきたしている

インクリメント
Libraryの調査と設定内容把握が終わりプロトタイプが動いた！

スクラムファシリテーター
POのプロダクトバックログに基づいて開発チームはスプリントバックログを作ってください！

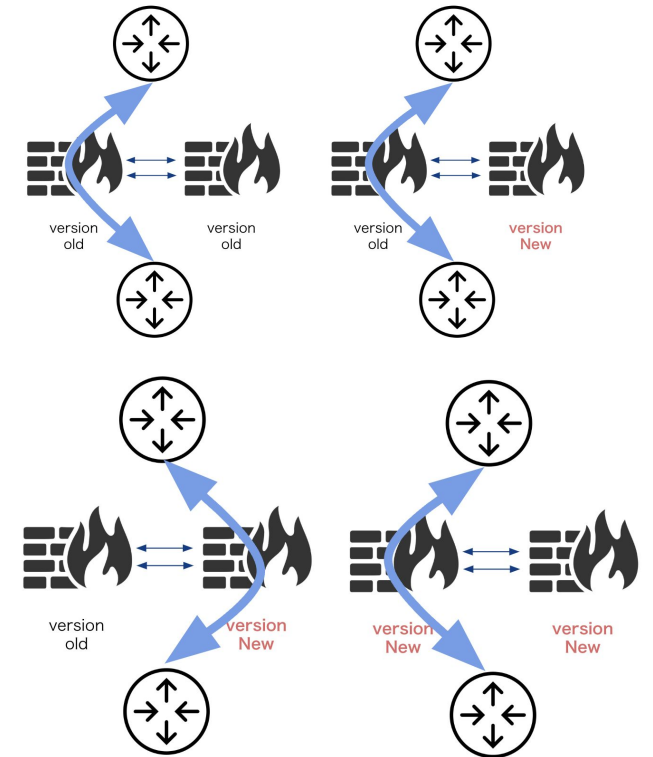
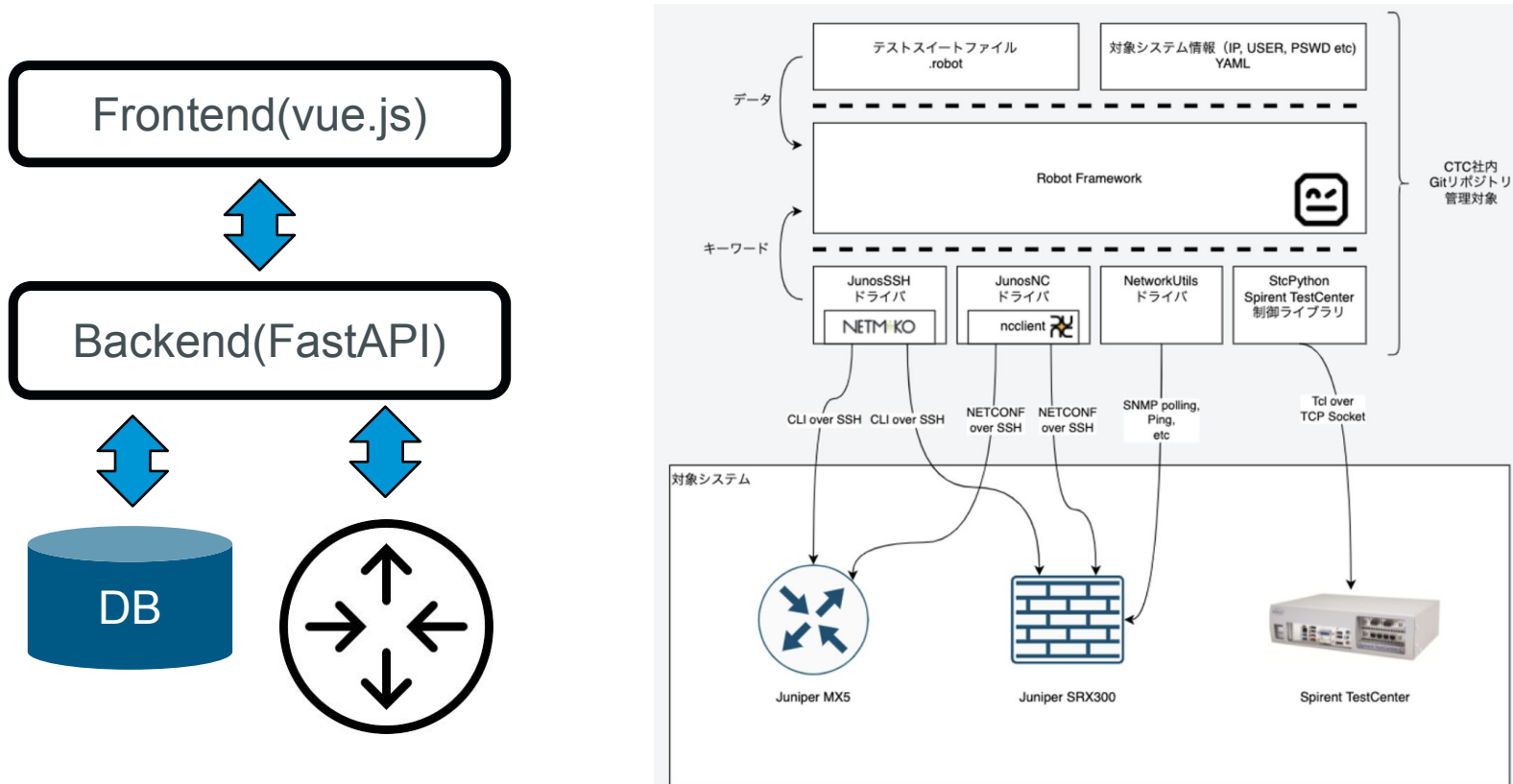


NW自動化チームが スクラムを採用した理由



NW自動化開発業務の分類(技術観点)

- Webオーケストレータ+NW装置モジュールの開発
- RobotFrameworkを用いた検証自動化の開発
- RobotFrameworkを用いた作業自動化の開発



受託開発で 開発するもの

- ・継続的SRE活動の開発
- ・ツールの一時的開発

技術の向上活動

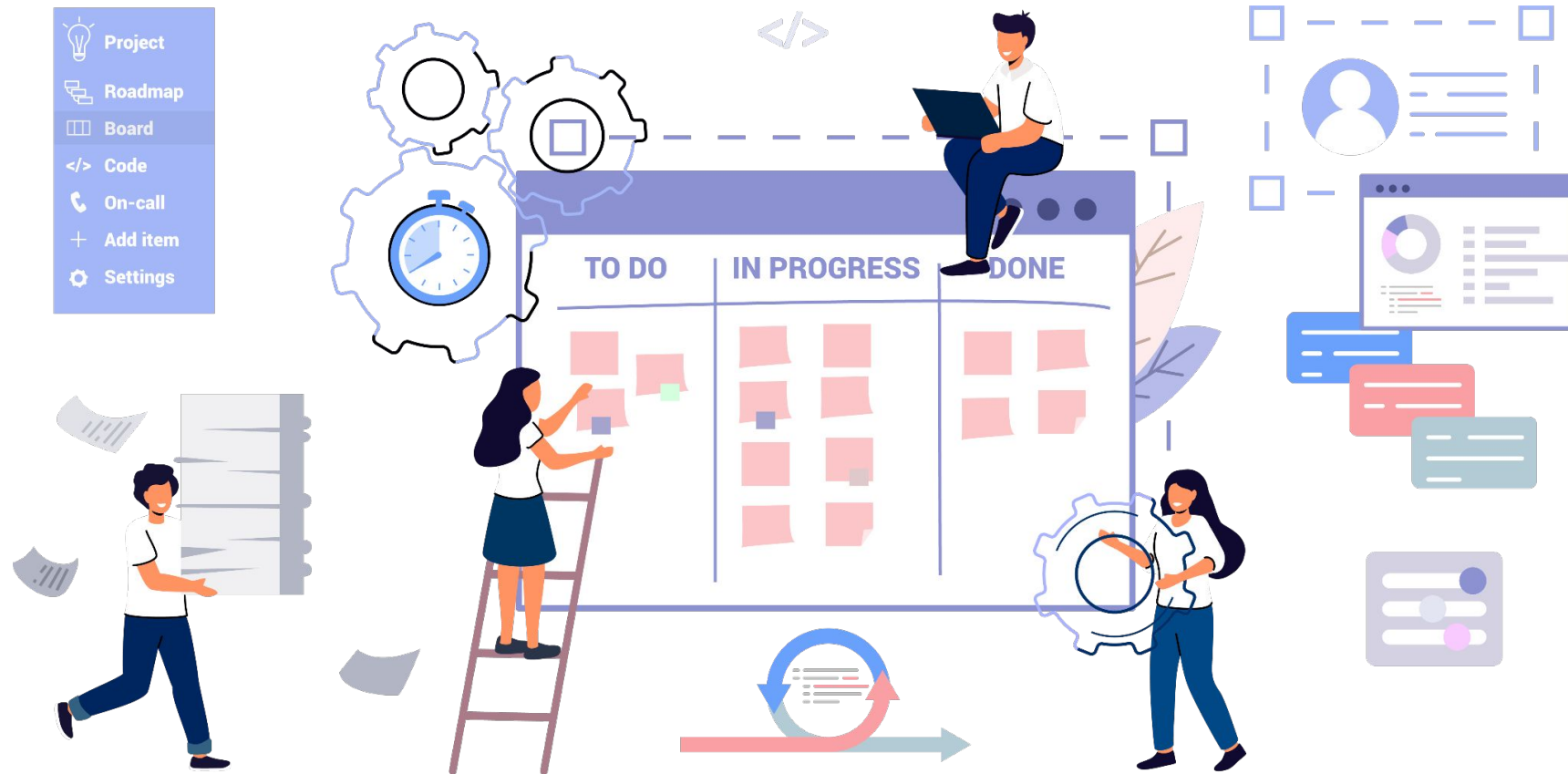
- ・新製品・技術の検証
- ・OSSへの貢献

社内コストでの 内部向け開発

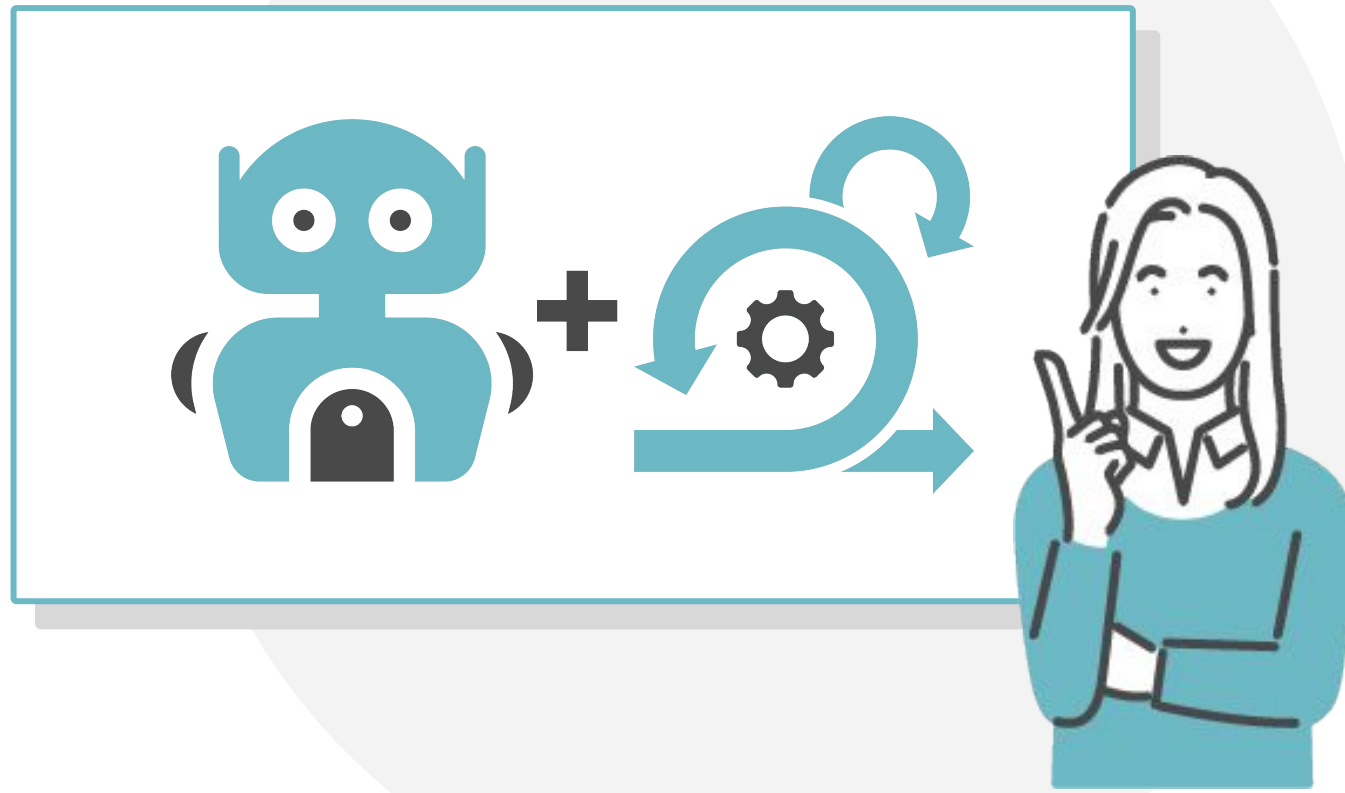
- ・RFP検証・出荷検査
- ・業務作業自動化のRPA



- 多岐にわたる技術・変化の激しい環境でアウトプットを出す必要があった
- Projectに1PMの様な従来のチーム編成では限界があった
- チーム運営手法としてスクラム開発を採用せざるを得なかった



NW自動化とスクラム開発の親和性

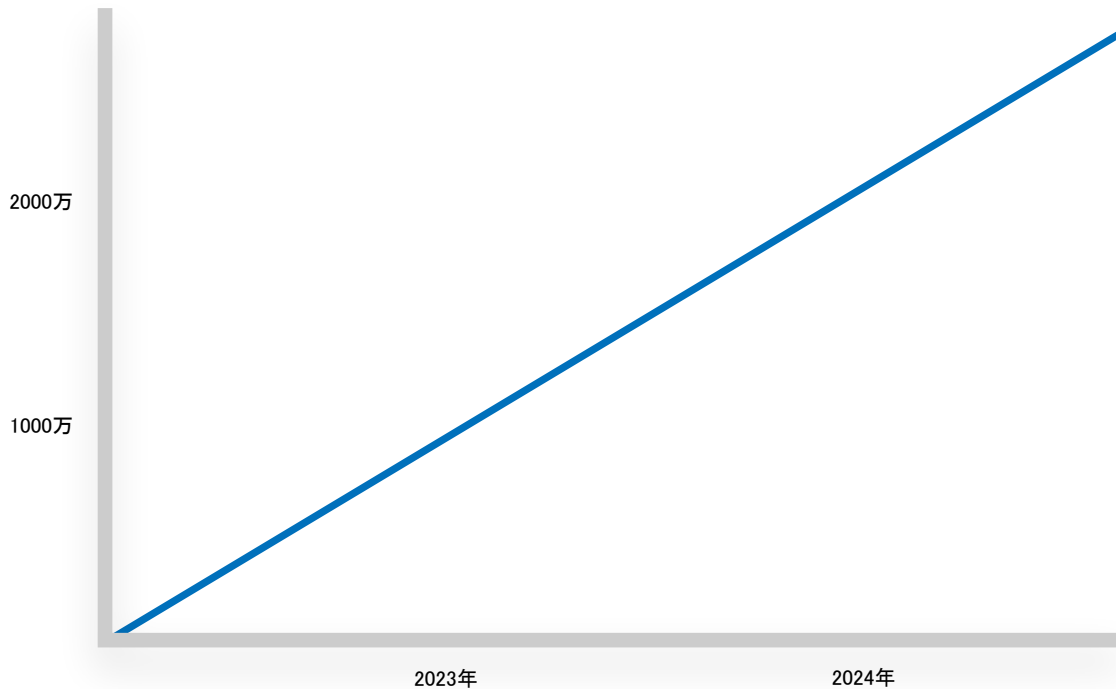


- NWの自動化は、あくまでもNWの作業を自動化したいだけなので、
主役であるNWの要件の変化に迅速に追従する必要がある

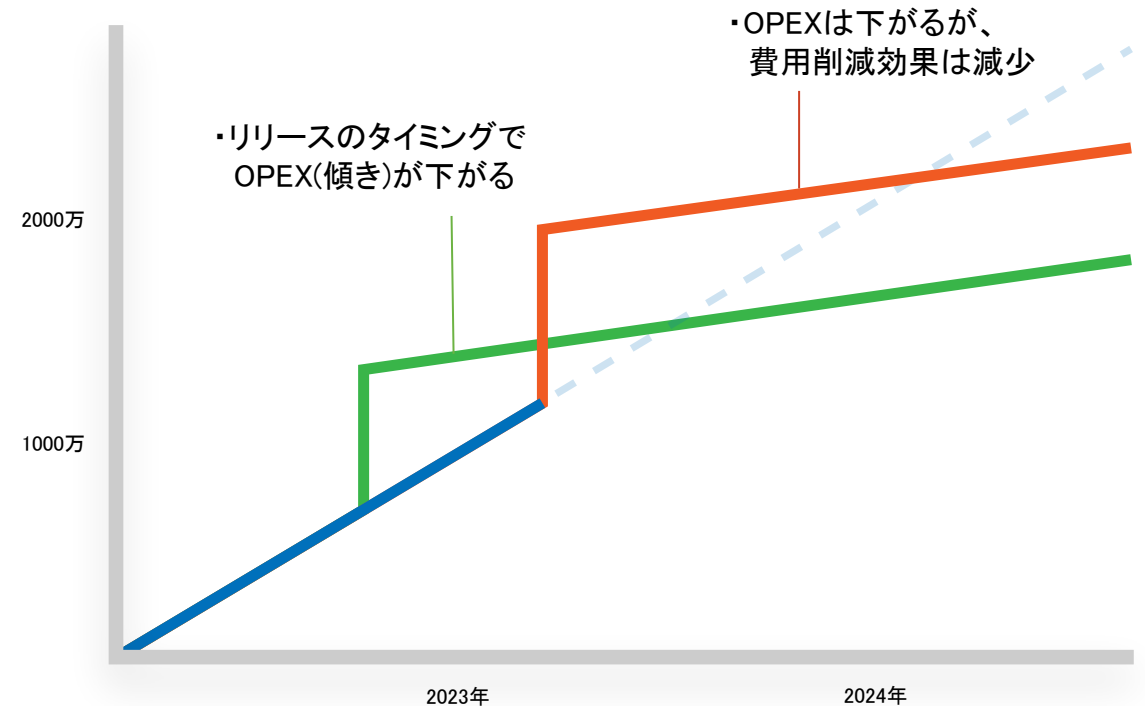


- 自動化によるOPEXの削減は、リリース後に得られた費用削減の累積となる。つまり、リリースを早くすることが重要である

手動運用の場合の手動でのコストの累積



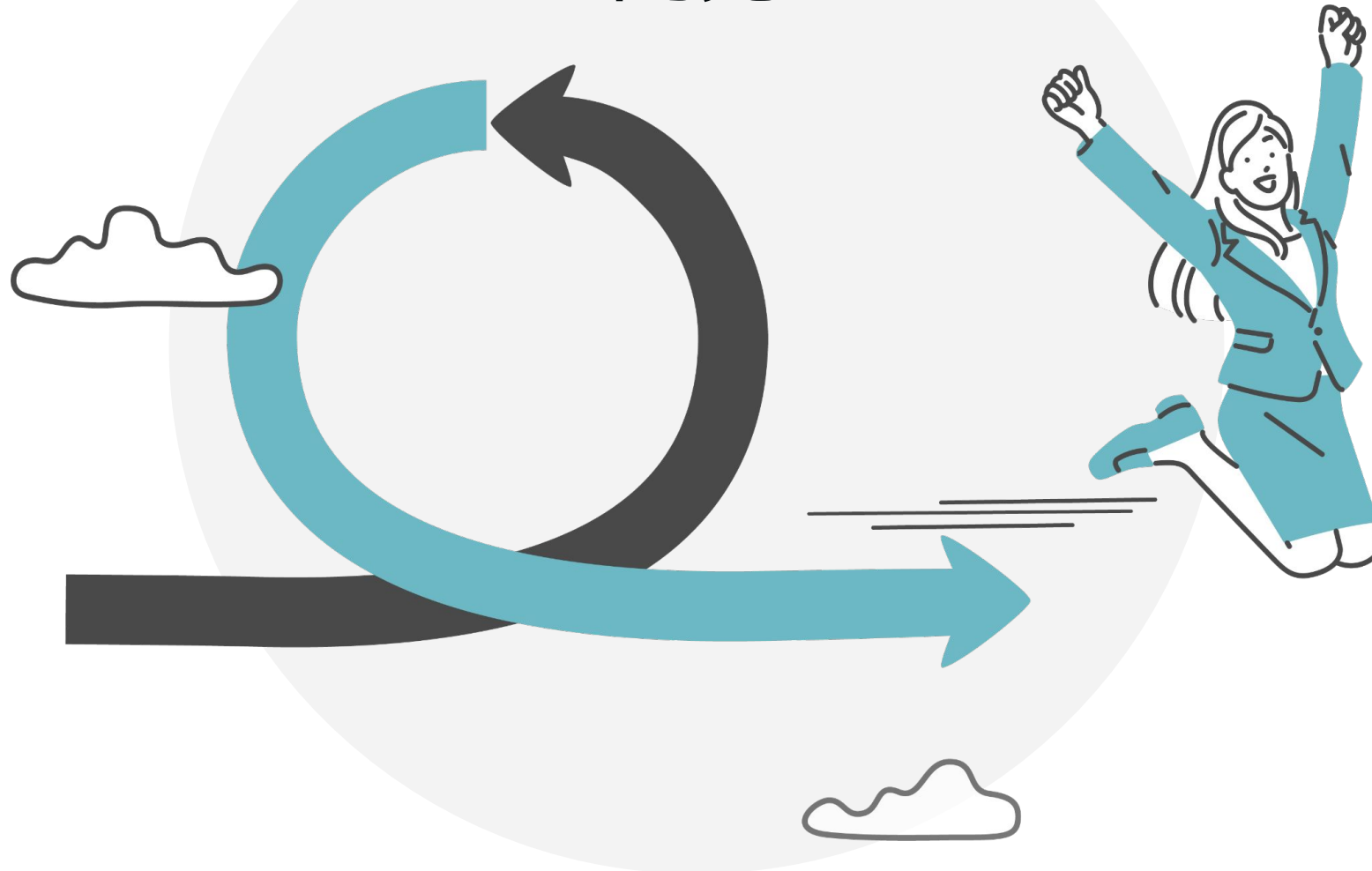
自動化した場合のコストの累積



- NWの技術・運用を理解した上で、開発技術にも長けているメンバーで構成したい。
必然的に少数精鋭になるため、効率の良い運営が必要になる



NW自動化チームの スクラム開発のリアル



- 1つの課で1つのスクラムチームとなっている
- メンバーは18名で構成されている

- PO : 1名
- CSM : 1名
- スクラムメンバー : 16名
- 社員 : BP 9:9 平均年齢 28歳



PO

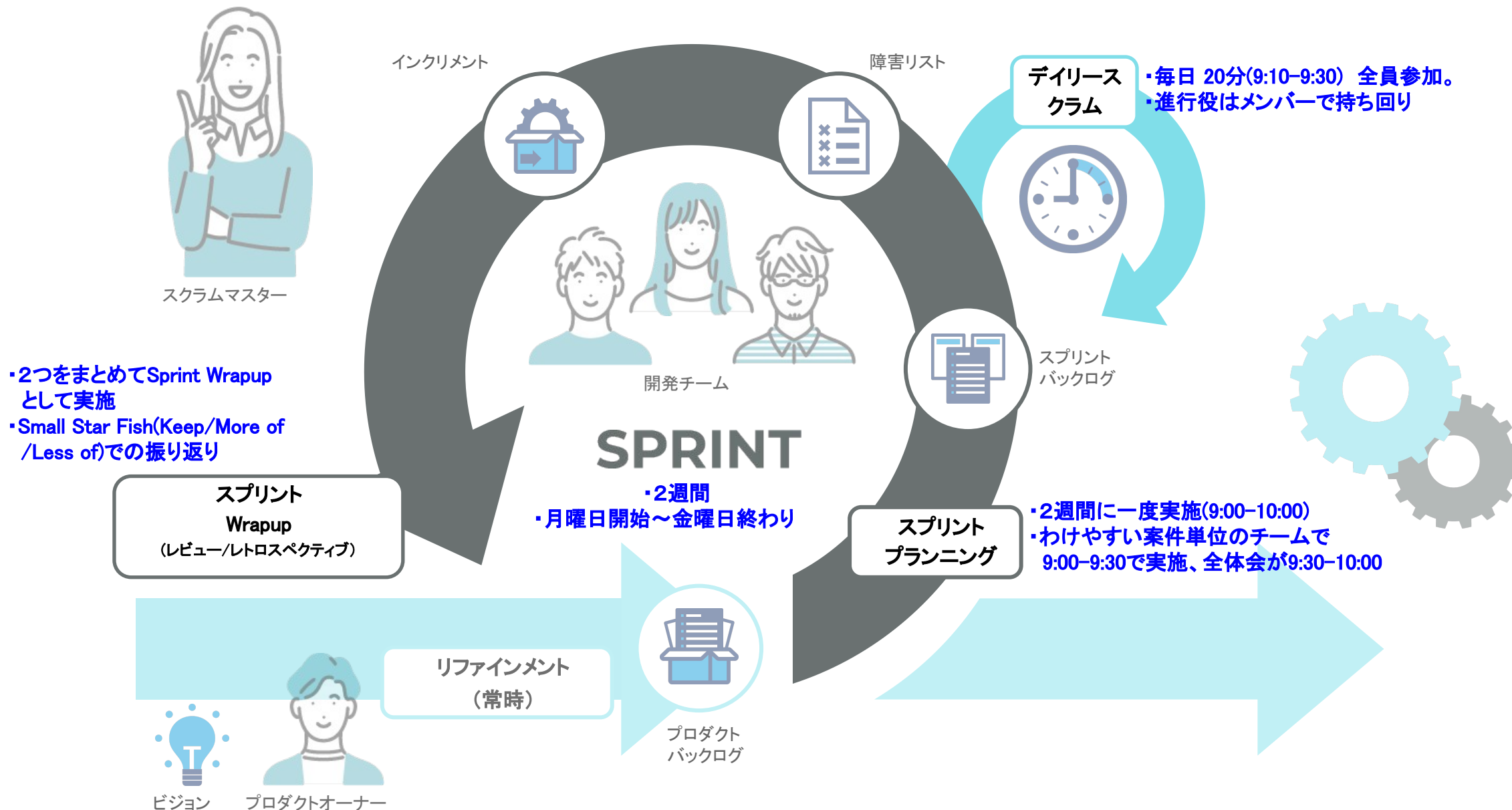


CSM

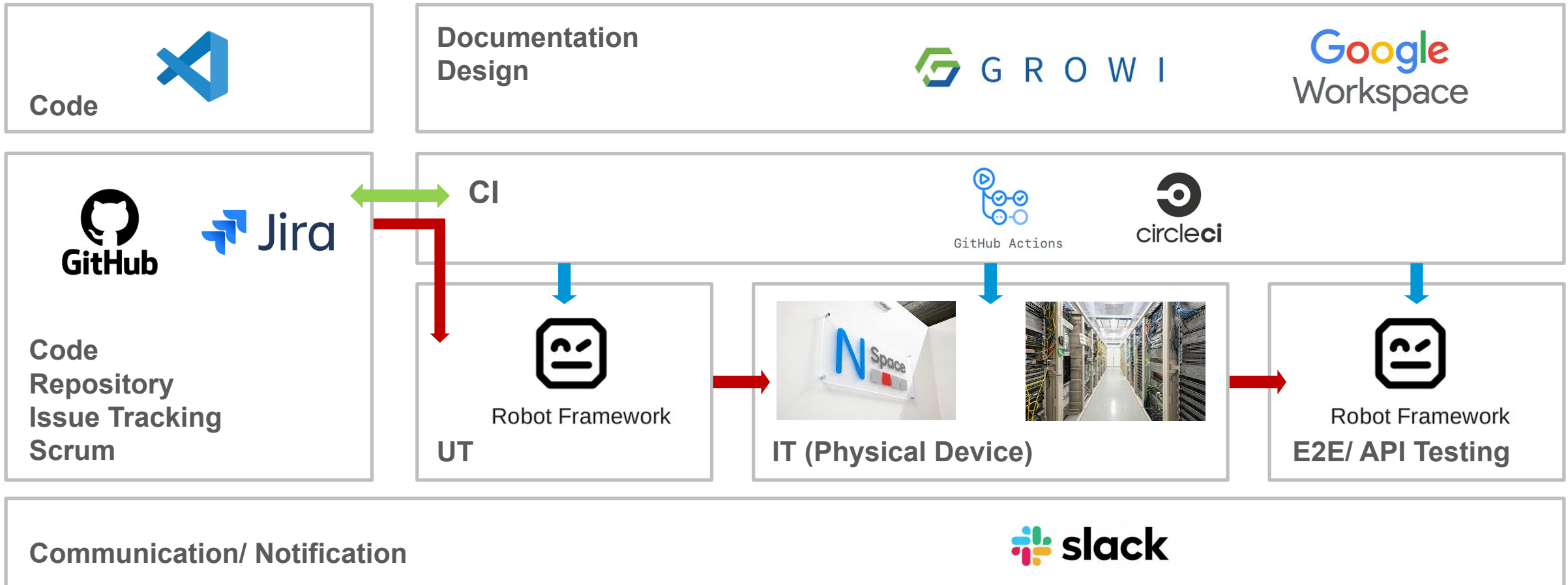


スクラムメンバー





- アジャイル/スクラムを実施するにあたって、適切なツールを使うことは最重要事項
 - JiraがマスターDBおよびInterfaceのような形になり、各種ツールと連携している
 - エクストリームプログラミングのプラクティスもペアプロ・テスト主導開発は取り入れている



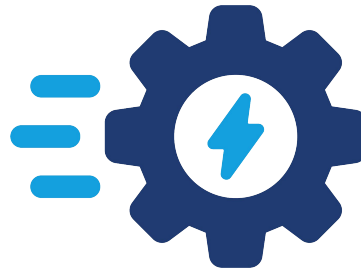
よくなったこと





DailyScrum

情報共有目的の会議は減り、
コミュニケーションの質と速度が
向上し、生産性も向上



スクラムボード(Jira)

実施内容/完了条件/状態が
明確化/共通化/可視化されたことで
手戻りや認識のずれが減少



レビュープロセス

完了条件のReviewや
ペアプログラミングの導入で
Reviewが細かい単位でスムーズに
行えるようになり品質が上がった



DailyScrum/ Planning/Wrapup

誰が何をやっているか
何に困って何を考えているかが
簡単にわかるようになった



DailyScrum/ スクラムボード(Jira)

常に他案件やタスクの状況が
入力されている状況になり
タスク/案件のアサインが楽



Scrum

会話や顔が見える機会が増え
やっている内容もわかるので
単純にみんな仲良くなった

悩んでいること



悩んでいること - 受託開発におけるスクラム開発



理想のスクラム開発

スクラム開発は基本的には内製でのソフトウェア開発を前提としている考え方



現実のスクラム開発

CTCでのスクラム開発は内製開発よりも受託開発が多いバランスになっている



解決策

スクラムの基本思想に基づいたスプリントごとにリリースのような形はできないが、スクラムの亜種とすることで恩恵を受けることは可能。



理想のスクラム開発

アジャイル開発は基本的には Team=Product(Service)がBasicな考え方になっている

また、チーム構造はメンバーはフラットな立場が理想の考え方



現実のスクラム開発

会社組織では1つの課=1スクラムにならざるを得ないケースが多い。課の中では複数のPJを並行するため1PJ=1スクラムは現実的ではない結果として、18名でのスクラムに。

スクラムチームはフラットであるべきなのだが、課長・部下という役割論も持ち込んでしまう



解決策

課としては1つのスクラムチームとし、PJの関連性があるメンバーでチーム化する。ただし、DailyScrum等のイベントはメンバー全員でやる

課長がスクラムマスターはおすすめしない。プロダクトオーナーまたはスクラムの外にいる形が理想。



理想のスクラム開発

タスクは開発メンバの誰にでも割り当てられる

それによって、チーム全員でタスクをこなせるから属人的も解消するし品質も上がる！



現実のスクラム開発

組織が技術縦割りの状況に加え専門外はわかりませんのスタンスの人も多い

さらに、IT業界の人手不足により未経験の人等も入ってくる

一騎当千みたいな人もいる



解決策

技術蓄積・伝承・育成の仕組みを整えることが大切

・情報の属人化をしない。蓄積と同時に伝承する文化を

・ペアプログラミングで、有識者が若手に効率的な知識伝承

・オンボーディングの整備で、育成が当然の仕組み・マインド作り

大事なこと





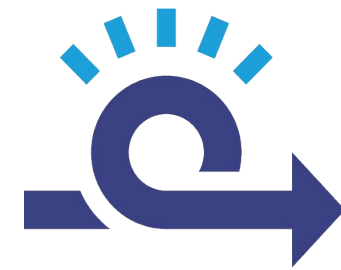
スクラム手法の正しい理解

正しいスクラムの手法および
ツールの使い方を
開発メンバー全員に浸透させる



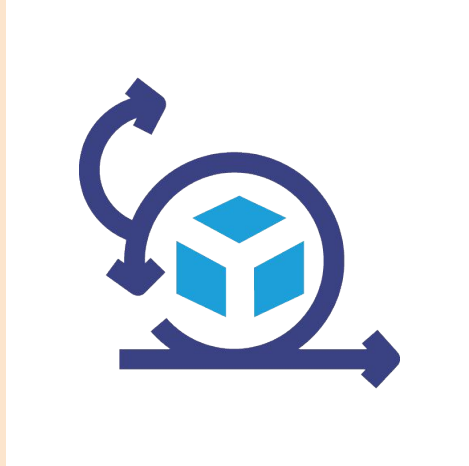
スクラムコンセプトの浸透

手法を理解するだけでは
ただのルールになってしまう。
コンセプトを浸透させることで
改善意識が生まれる真のスクラムへ



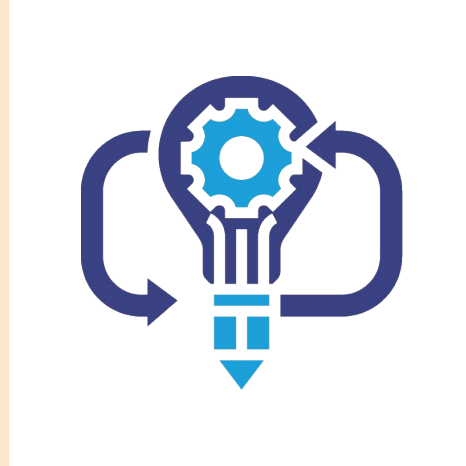
Why Scrum

そもそもなぜそんなことが必要？
仕事・技術とは何か？
生きるとは何か？



Productの理解

チームにおけるProductとは何か
実施している作業は何につながり
何を価値としているか



Vision/Roadmapの理解

Productの先にあるのは何か
VisionとRoadmapは
果たして何を指しているのか
自分の思想・成長とリンクするか



組織目標/社会課題

VisionやRoadmapの先には
組織の目標(責任)や社会課題がある



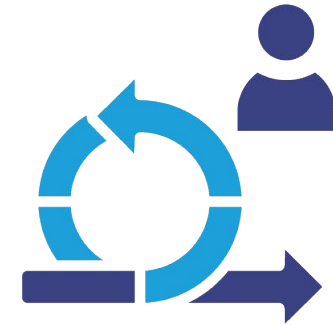
毎日のファシリテーション

自身がファシリテートをする時はもちろんのこと周りのメンバーがしっかりとスクラムプロセスを実施できることを浸透させる



チーム改善活動

チームとしての改善活動の歩みを止めてはならない課題意識と献身的なマインドの両立が必要になる



サーバントリーダー

従来型の「俺について来い」のリーダーではなく視野を広く一歩引いた目線から全体最適を考えられるリーダー要素

- ScrumTeamMember + α
 - NT課(CTC NW自動化開発チーム)メンバーの皆様+チームに関わっている社員の皆様
- Customer
 - 日々案件を頂ける皆様
- Designer
 - 水野修さん
- JANOGer + 運営の皆様

- アジャイル/スクラムやってますか？
- よかったこと・悩んでいることありますか？
- その他、なんでも！

CTC

▼ *Challenging Tomorrow's Changes*