

大規模スクラムの失敗から学んだこと

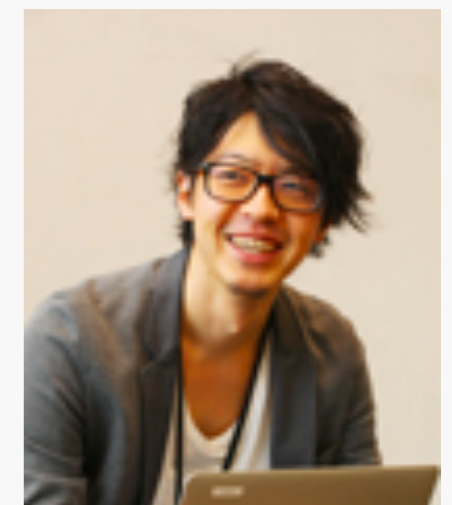
～急成長をとげるAirレジの組織拡大の取り組み～

(株)リクルートライフスタイル 塚越 啓介・佐橘 一旗

自己紹介



- Itsuki Sakitsu - エンジニア
 - 初期からのスクラム導入推進役
- Keisuke Tsukagoshi - エンジニア
 - スクラム支援、コーチング



大規模スクラムって？

普通のスクラム



Product Owner



Team



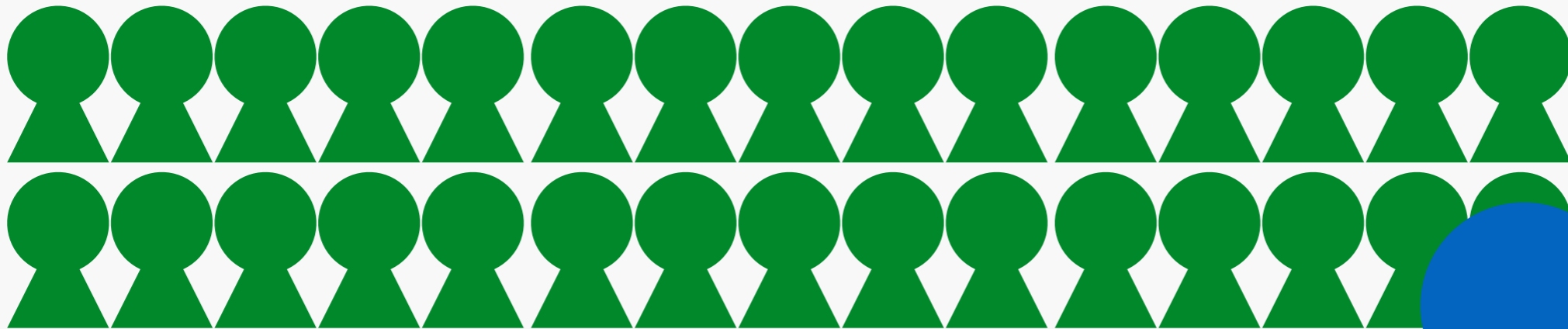
Scrum Master

Product

大規模スクラム？



Product Owner



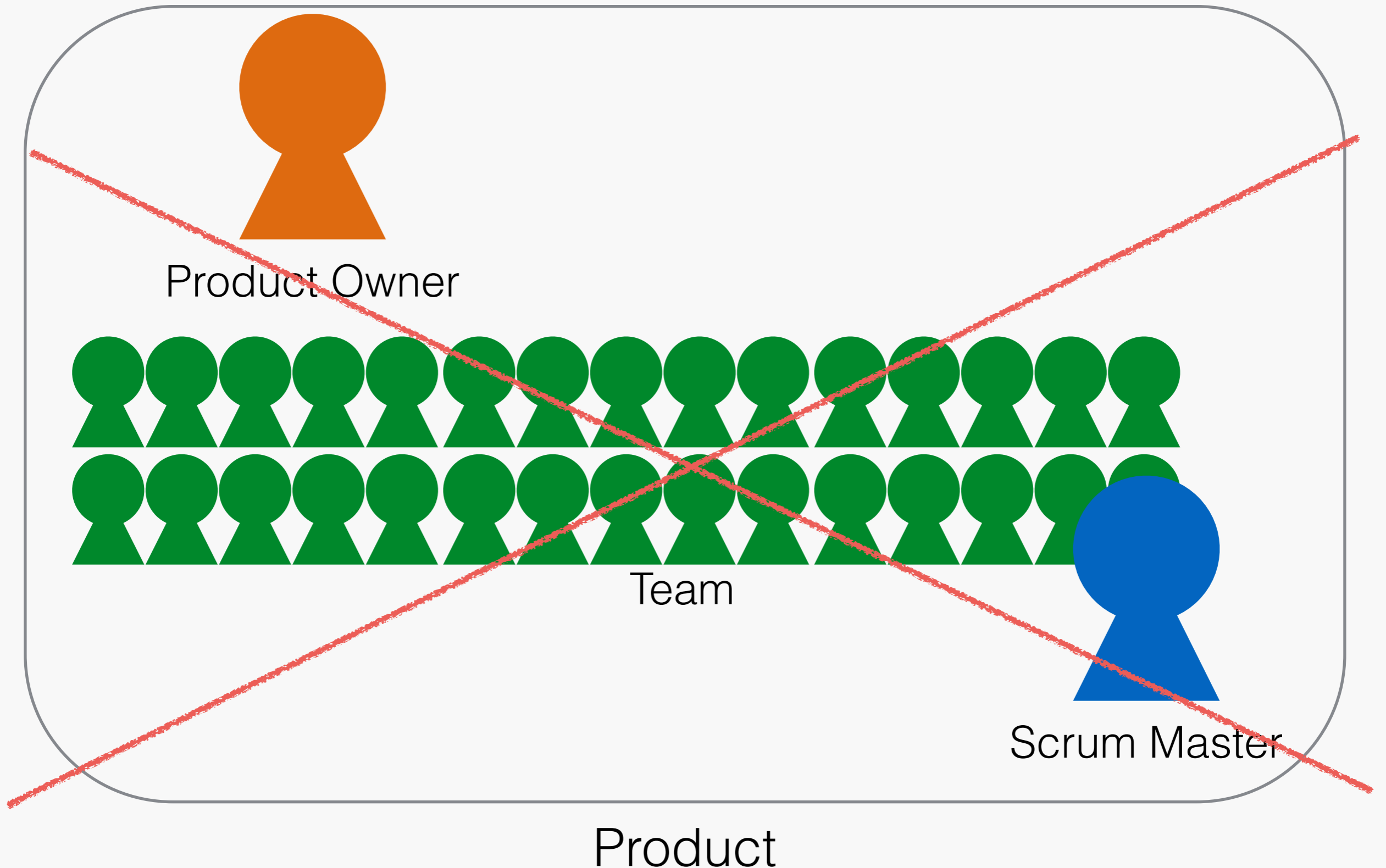
Team



Scrum Master

Product

大規模スクラム？



大規模スクラム



Company Board



UX Specialist



Portfolio Architect



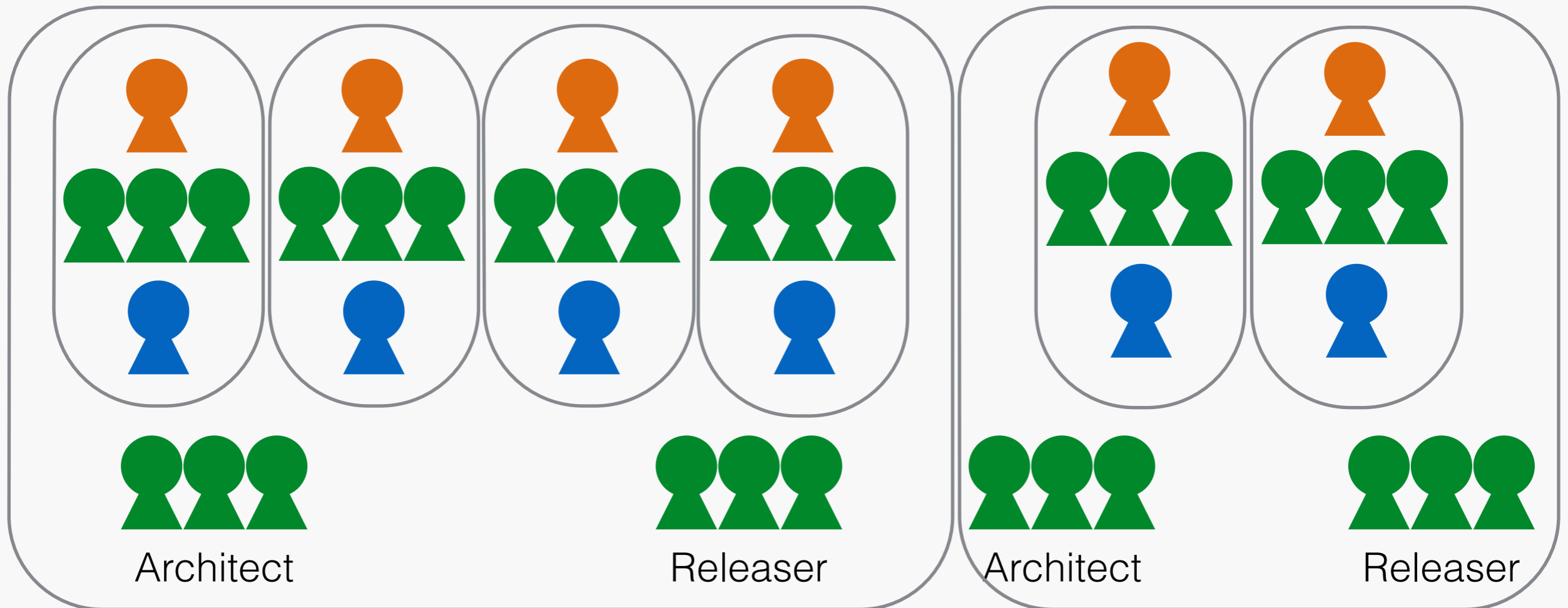
Scrum Support



PO Lead



PO Lead



Product

Product

本日お伝えしたいこと

スケールアップしても

スクラムの原理原則はかわらない

スクラムの導入に最も重要なのは

「Why」 「検証と適応と透明性」

アジェンダ

1. Airレジとは？
2. なぜスクラムに取り組んだのか？
3. 大規模アジャイル開発、検証と適応からの学び
4. 今後、挑戦したい課題
5. まとめ

1. Airレジとは？

About AirREGI

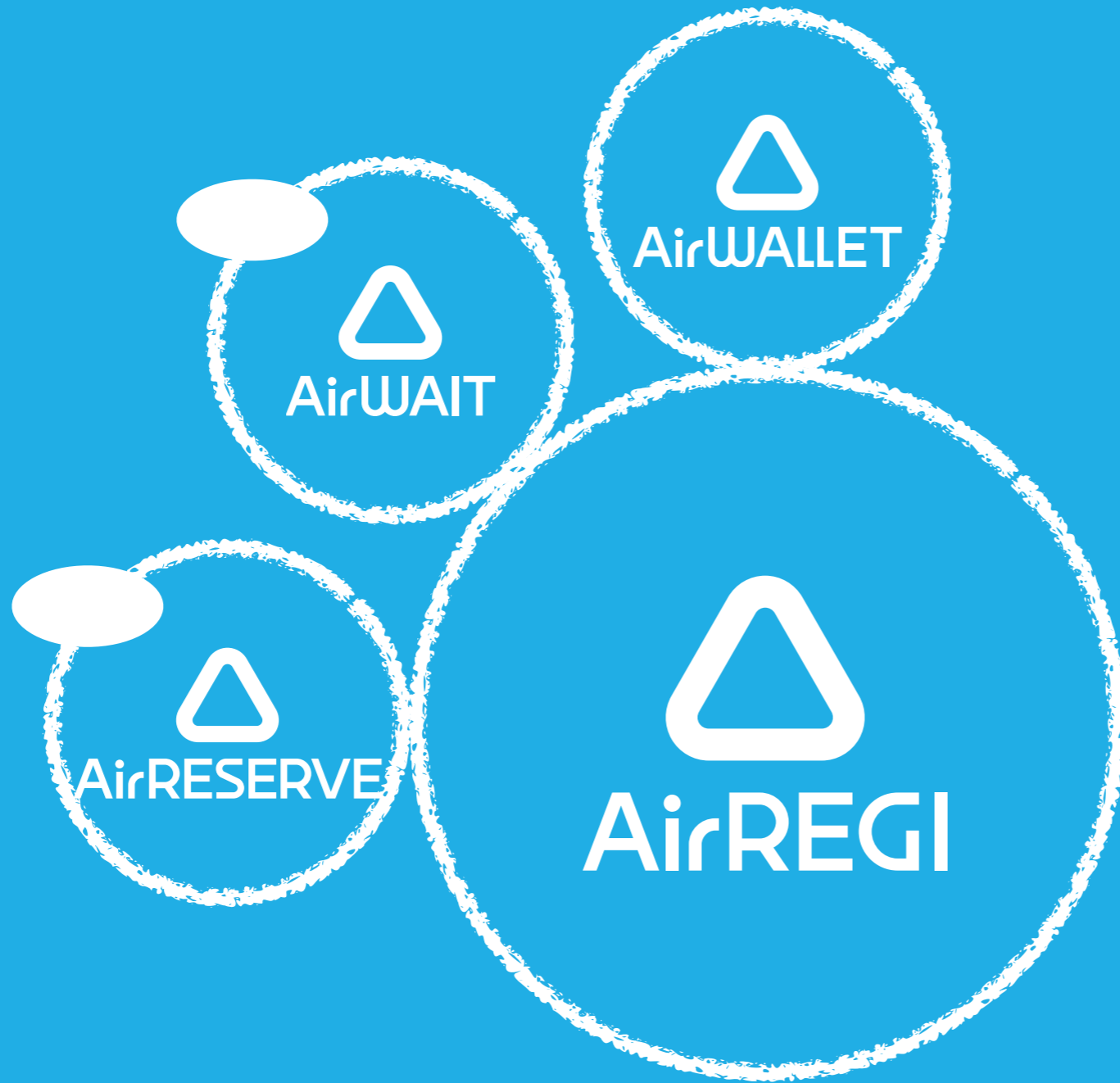
無料で簡単に使えるPOSレジアプリ

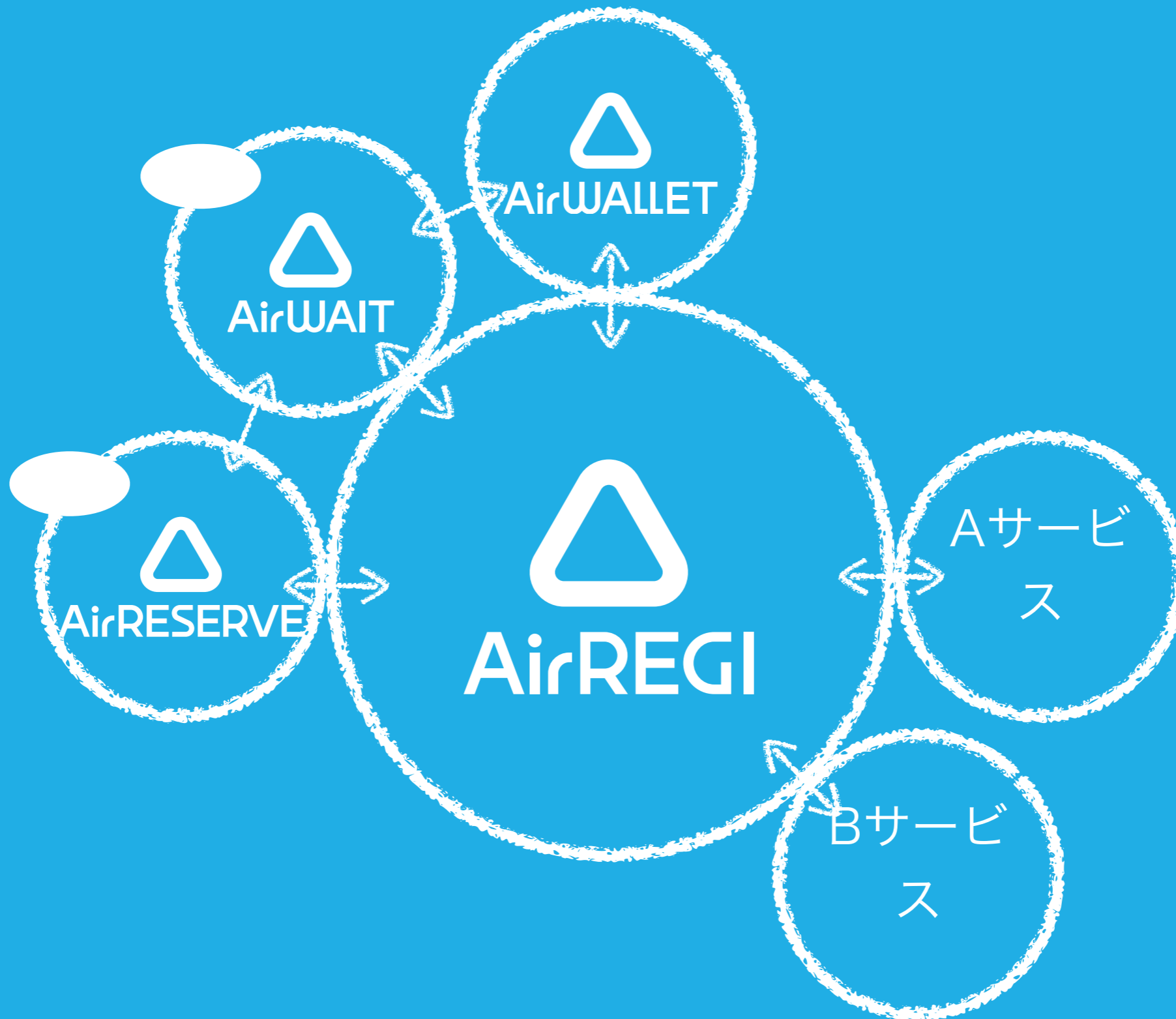
 **AirREGI**



GOOD DESIGN AWARD
2014年度受賞









我々は何故、

- **スクラム**を行うのか？
 - 未経験の事業領域、試行錯誤の必要性
- **スケール**する必要があったのか？
 - 相互に連携する複数のサービスを、全体整合性を保ちつつも並行して立ち上げたい

2.なぜスクラムに取り組んだのか？

Why SCRUM?

ぶっちやけた話

当たり前を当たり前に
やりたいから

当時の状況

要件決まってるから
開発できない



当時の状況

要件決まってるないか
開発できない

これ、全然イケテナイ
言われたから作るか



当時の状況

そんな話きいてないけど



当時の状況

こんなスケジュール無理
にきまってるじゃん



当時の状況

こんなスケジュール無理
にきまってるじゃん



目指したもの

- 率直な意見を言い合える文化
- チームが協力してものづくりを行える文化
- 常に成長を続けられる文化

なぜスクラムをはじめたのか

**当たり前前を当たり前前
やりたいから**

3. 大規模アジャイル開発、検証と適応からの学び

How we learned, tried and adopted

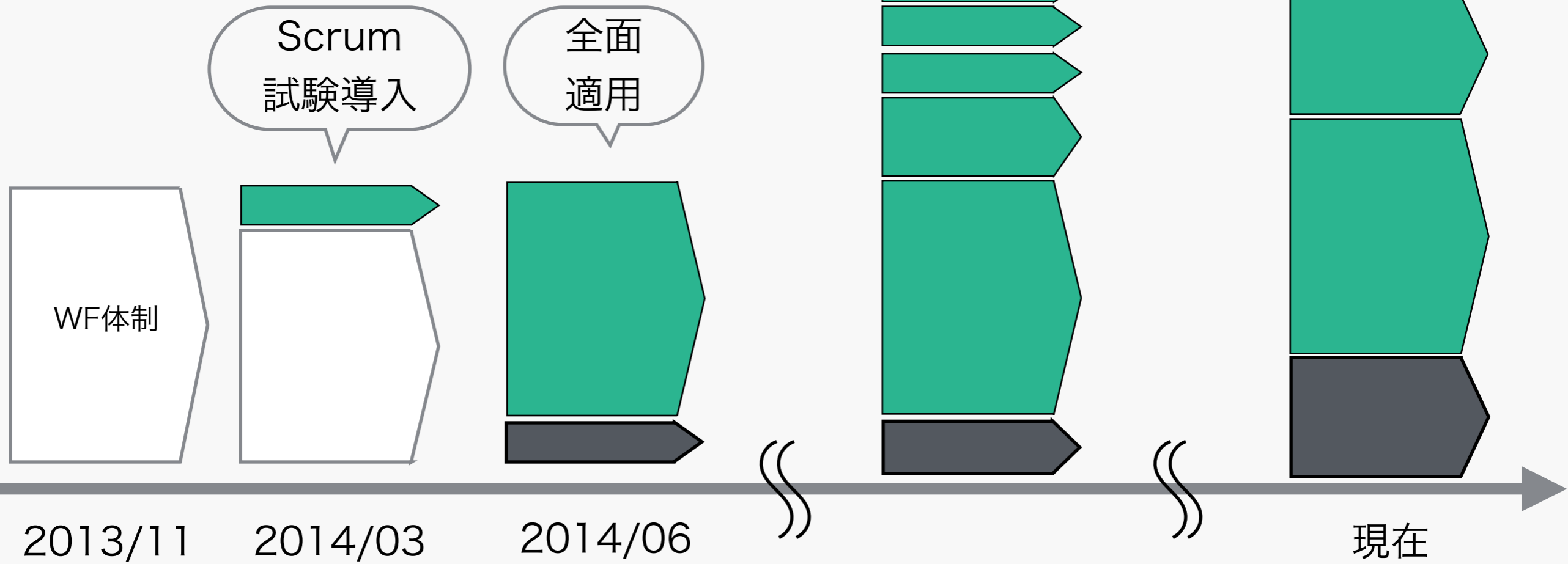
組織拡大の歴史

さらに拡大

Product / Component

既存体制

機能組織



Phase 1. Scrum試験導入

Product / Component

既存体制

機能組織

Scrum
試験導入

全面
適用

組織拡大

さらに拡大

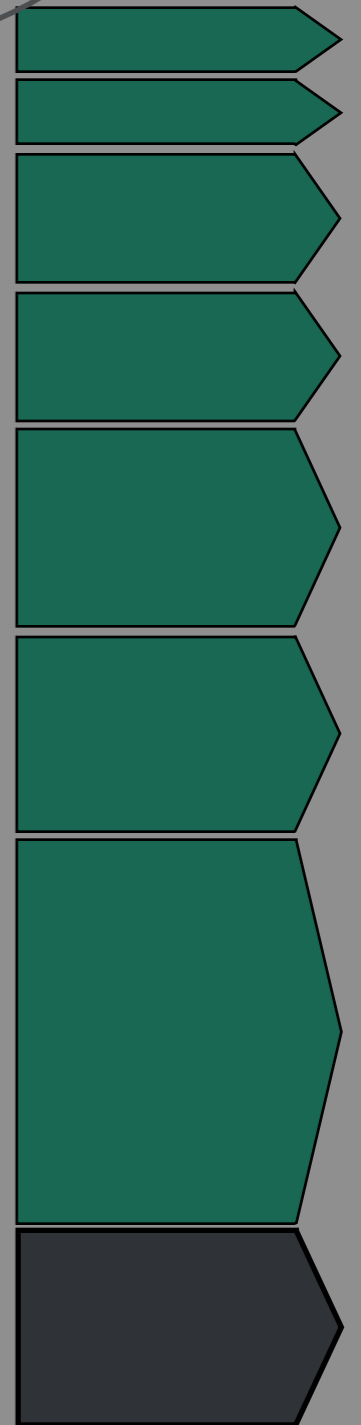
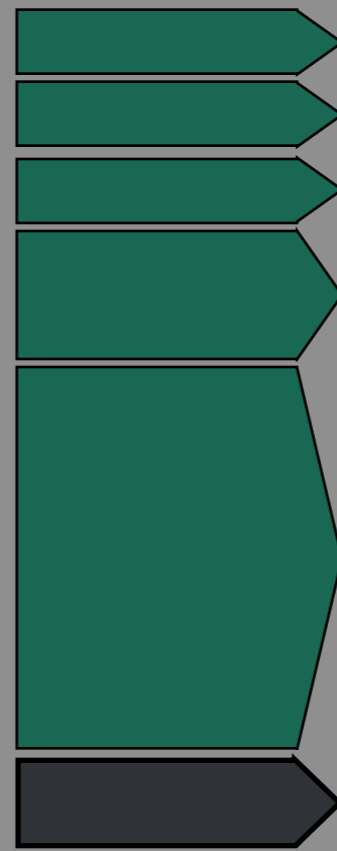
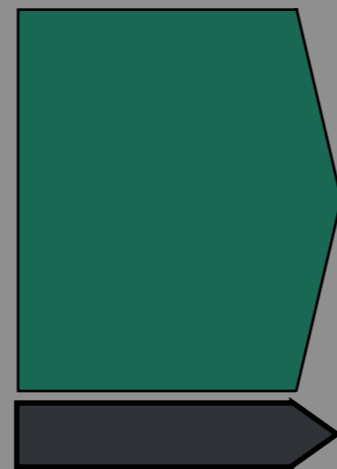
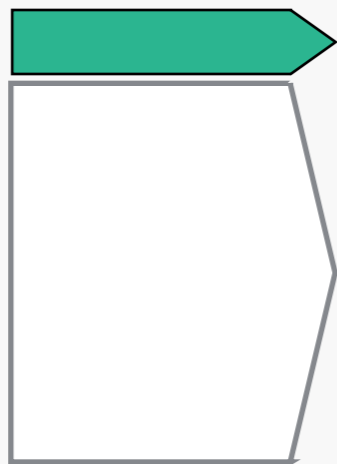
WF体制

2013/11

2014/03

2014/06

現在



糸

Phase2. Scrum全面適応

さらに拡大

Product / Component

既存体制

機能組織

Scrum
試験導入

全面
適用

組織拡大

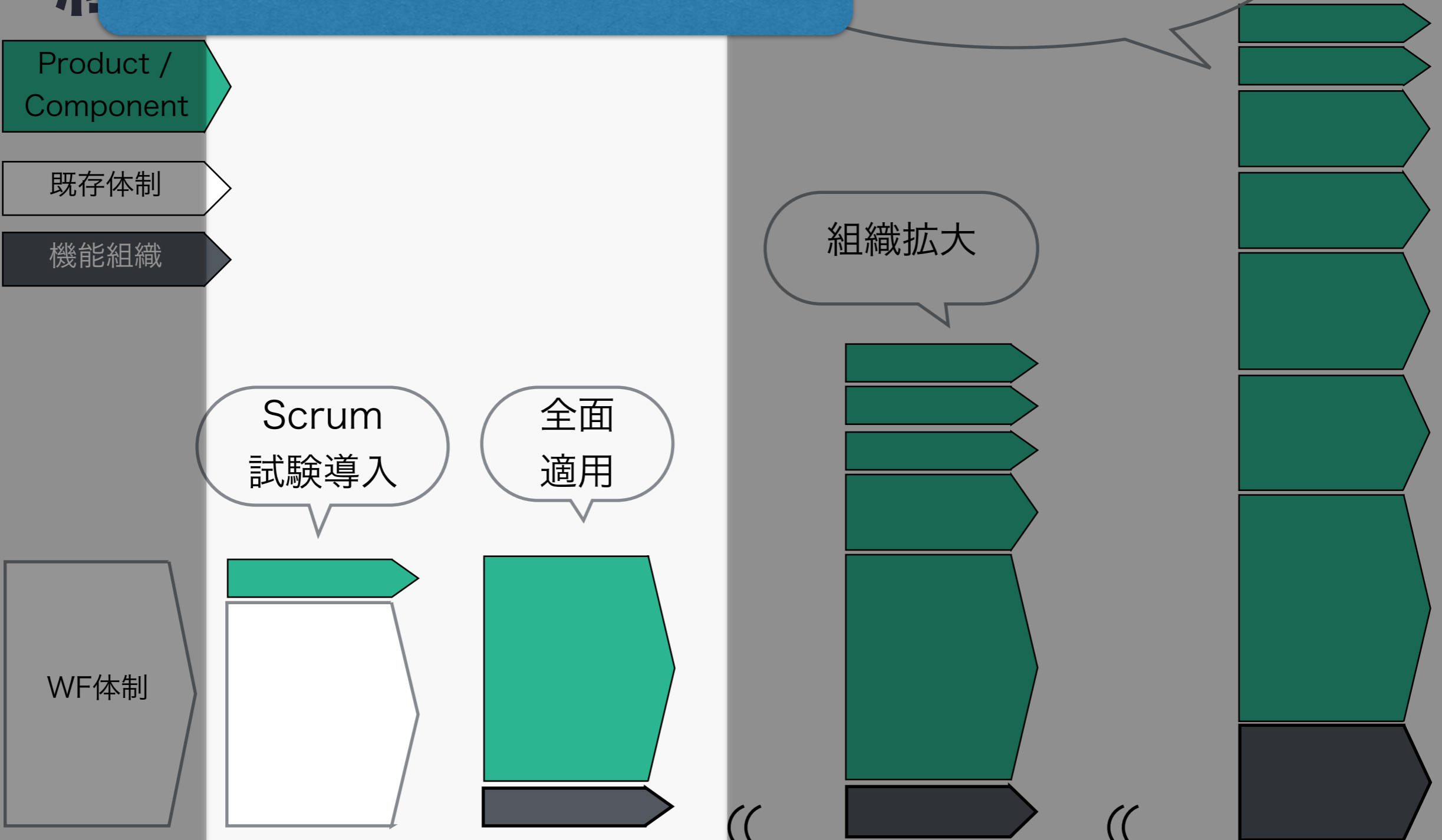
WF体制

2013/11

2014/03

2014/06

現在



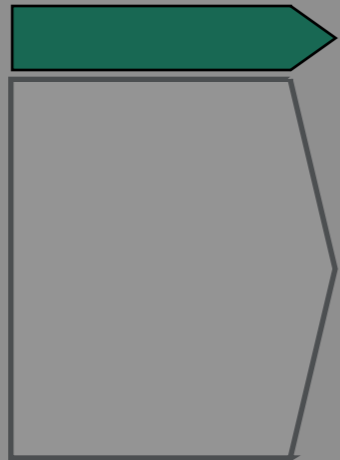
組織拡大

Product / Component

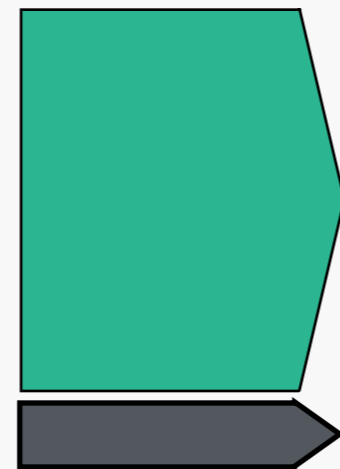
既存体制

機能組織

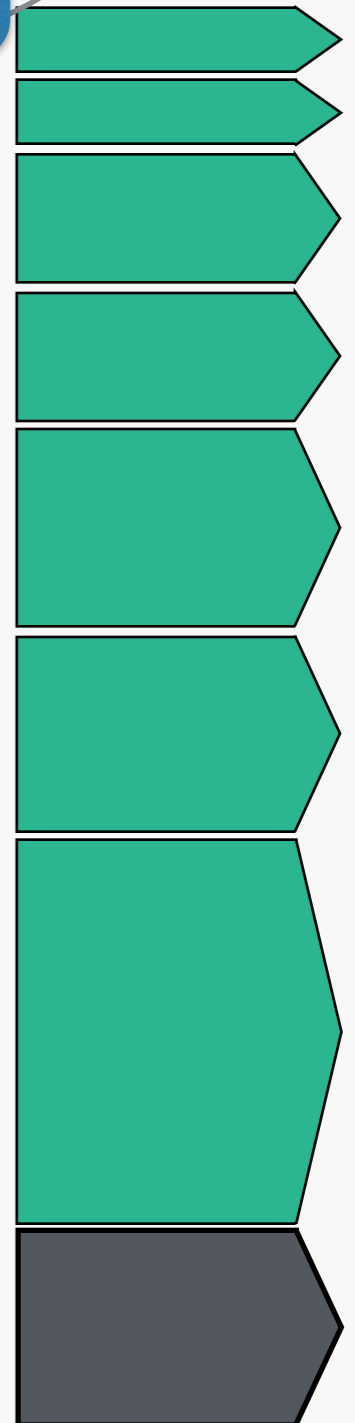
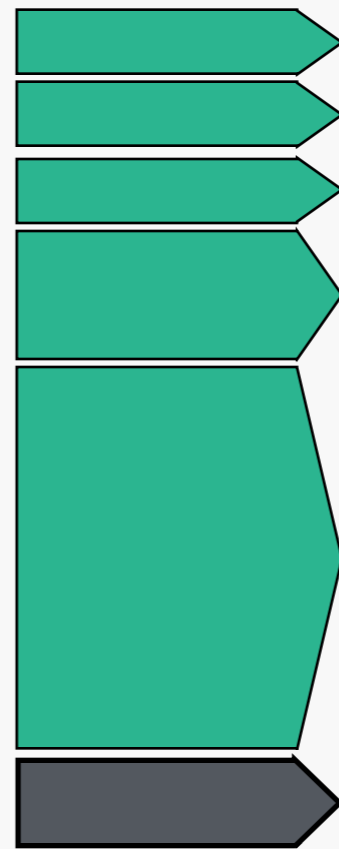
Scrum
試験導入



全面
適用



組織拡大



Phase3. 組織の拡大

2013/11

2014/03

2014/06

現在

2.1. WFからスクラムへ

How we started SCRUM

Phase 1. Scrum試験導入

Product /
Component

既存体制

機能組織

Scrum
試験導入

WF体制

全面
適用

組織拡大

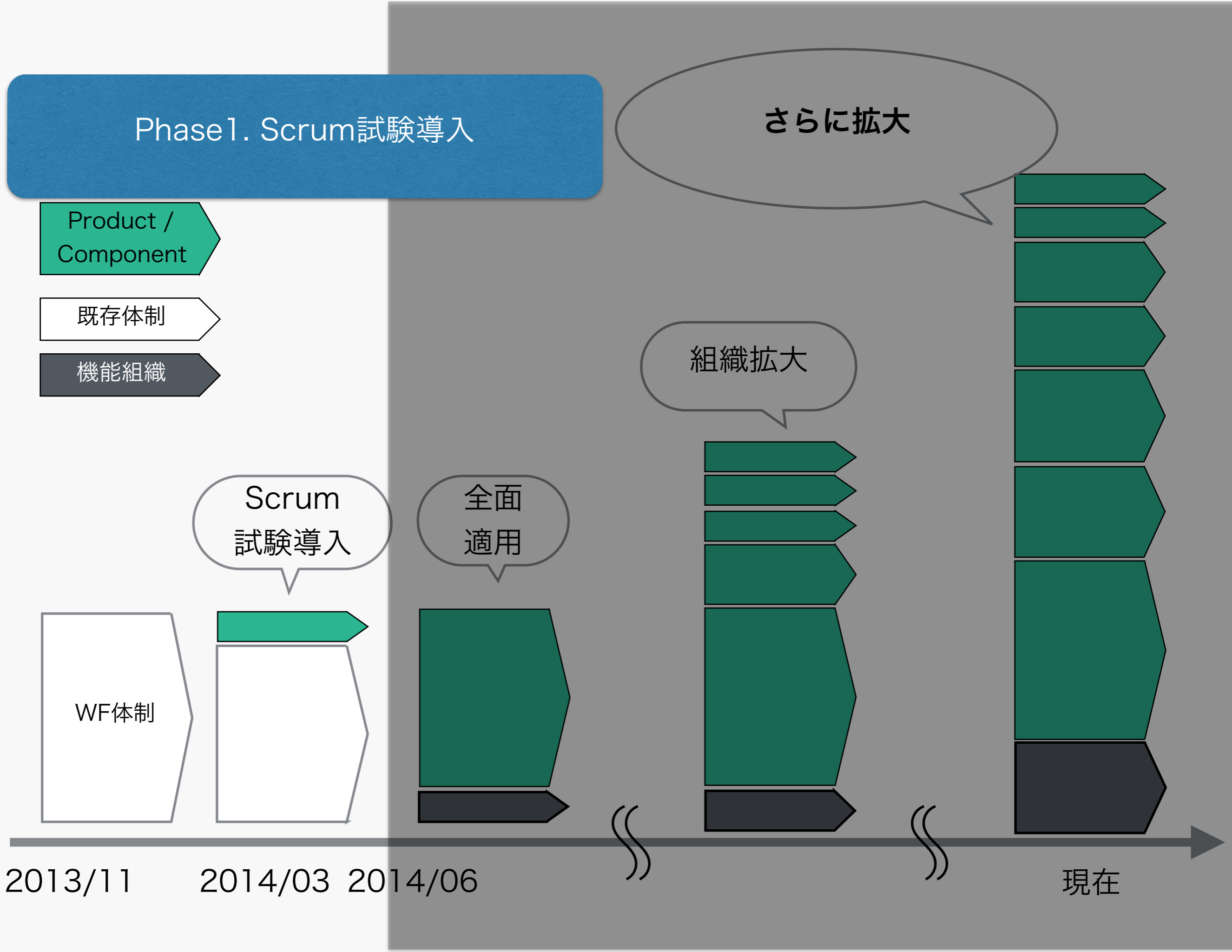
さらに拡大

2013/11

2014/03

2014/06

現在

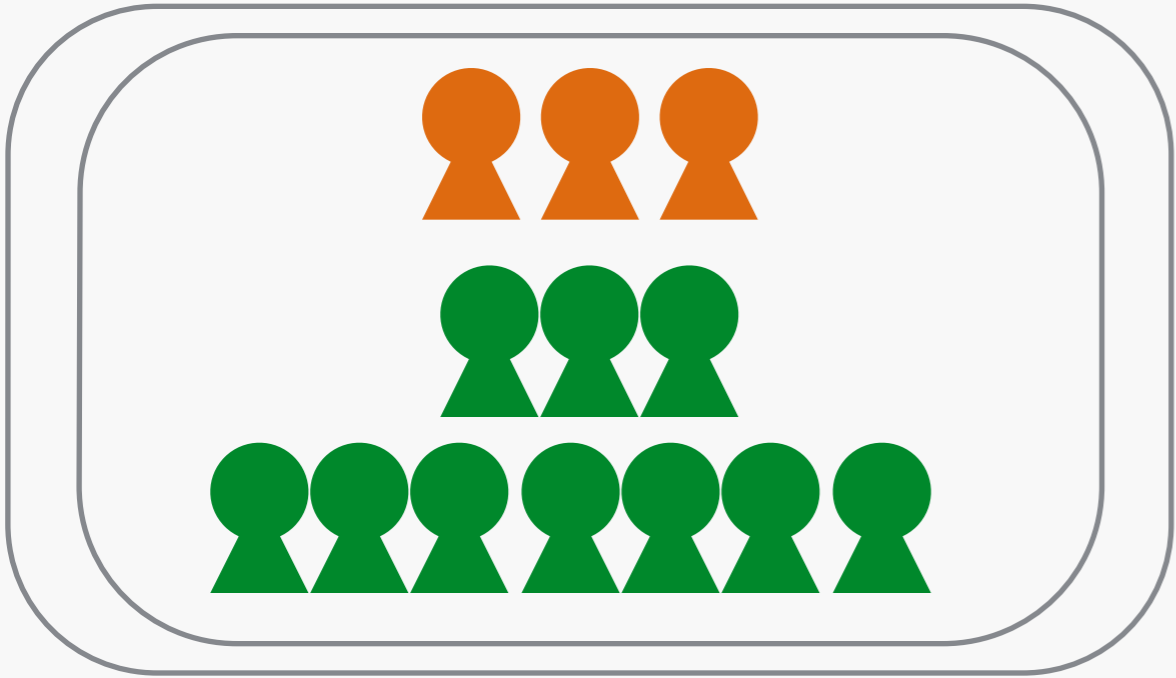


はじまり

- 「今のプロセスを改善したい」
- Scrum Boot Campを購入
- まずはカタチから

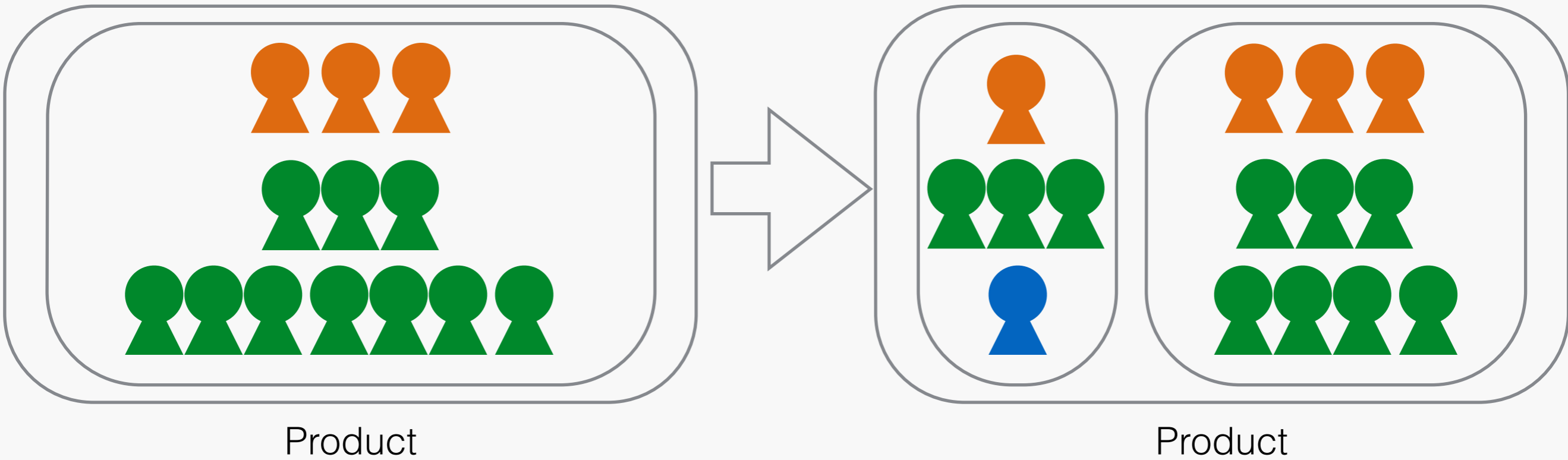


WF開発



Product

WF開発 + スクラムチーム



当初導入できたこと

- アーティファクト
 - プロダクトバックログ、スプリントバックログ(KANBAN)
- セレモニー
 - 計画ミーティング、朝会、振り返り、スプリントレビュー

やってみた結果

- 開発タスクが漏れなくなった
- 他メンバーが何をしているか見えるようになった
- 案件や進め方に対して意見が出せるように

スクラムよかった！

- 知見者がいない状態でも導入効果はある
- 形だけの導入でも効果はでる

スゲーうまくいった

スクラムうまくいったから
他のチームも
全部スクラムでやろう！

2.2. スクラムの全面適応

How we adapted

糸

Phase2. Scrum全面適応

さらに拡大

Product / Component

既存体制

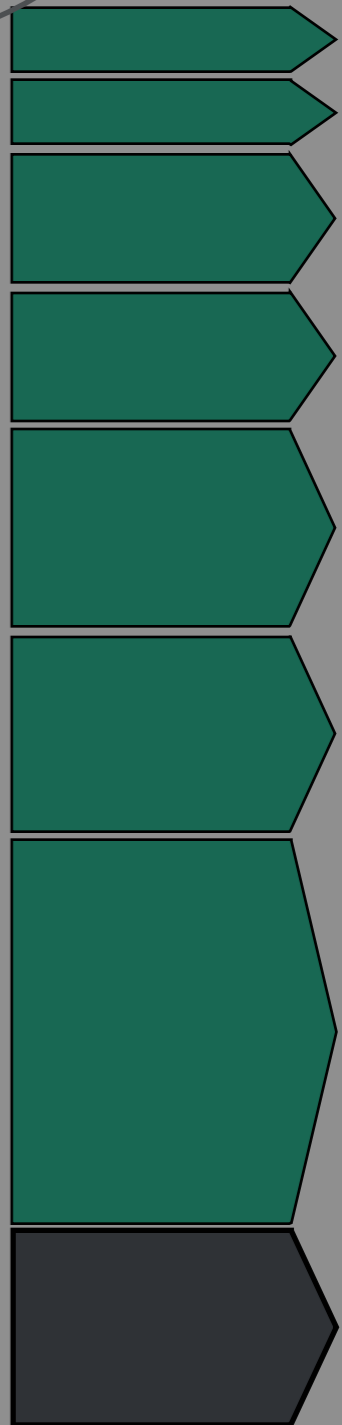
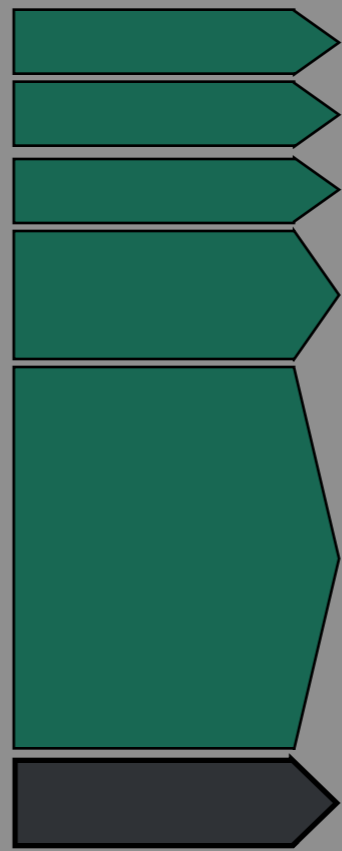
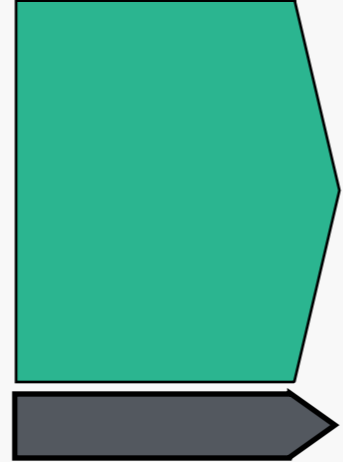
機能組織

Scrum
試験導入

全面
適用

組織拡大

WF体制



2013/11

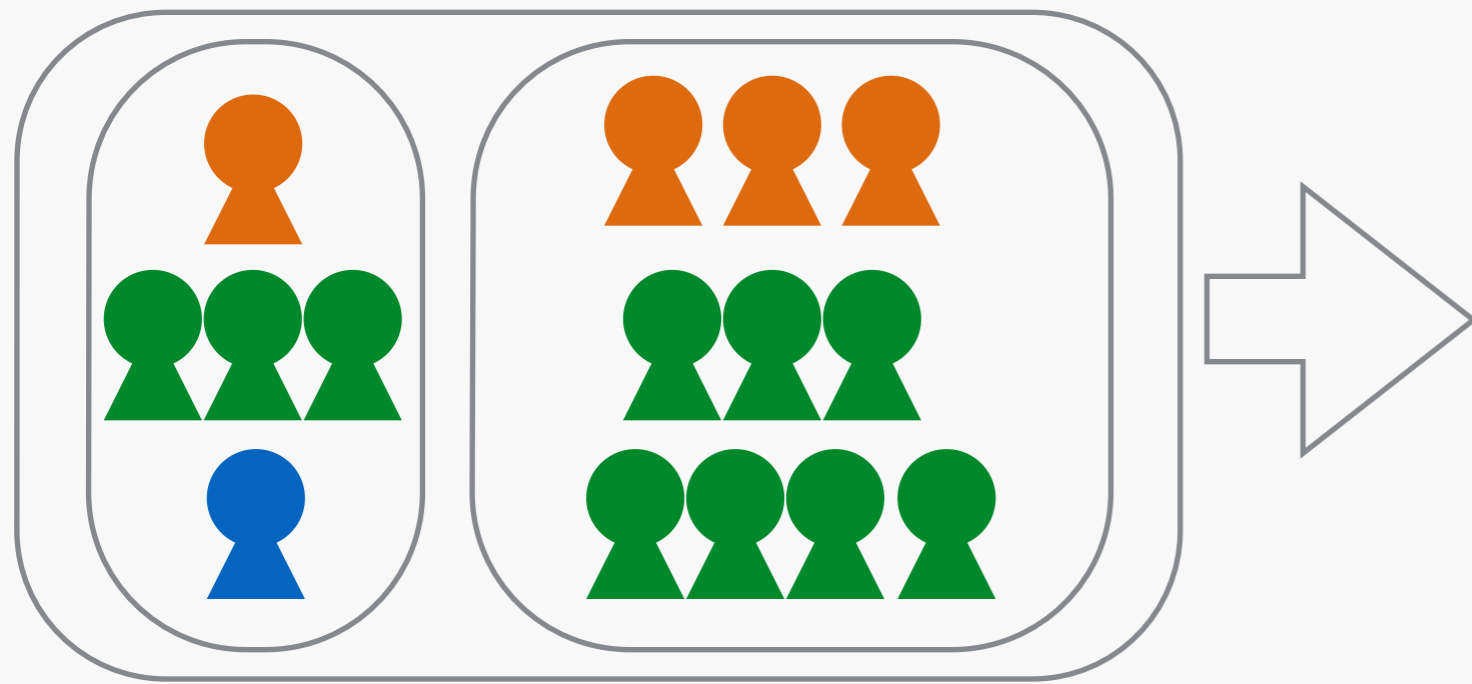
2014/03

2014/06

現在

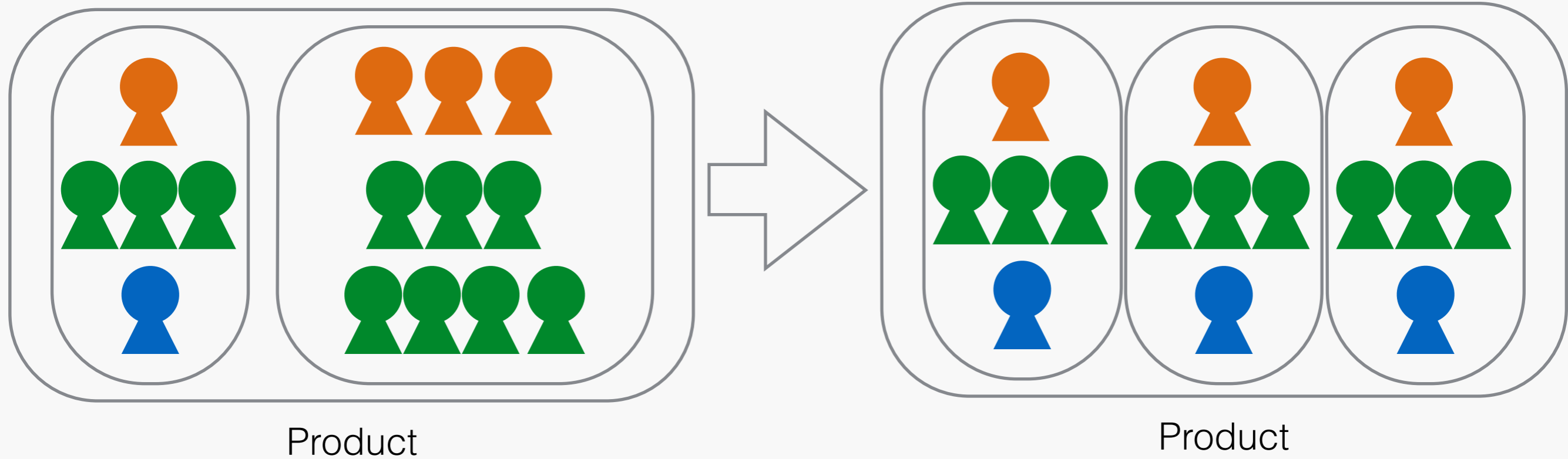


スクラム+WFチーム



Product

全面スクラム化



何が起こったか

SMってタスクの管理する人でしょ！

ユーザーストーリー作ったからあとよろしく！

スクラムだから納期はコミットしないでしょ？

あれ？

なんか違う

スクラムよかった！

- 知見者がいない状態でも導入効果はある
- 形だけの導入でも効果はある

正しくは

スクラムよかった

- 知見者がいない状態でも導入効果はある
- 形だけの導入でも効果はある

ただし、チームメンバーの多くが能動的かつ課題感を共有している場合に限る

落とし穴

- **トップダウン導入による、やらされスクラム**
- **形だけの導入によるスクラムの誤解の顕在化**
- **不安感からくる手段の目的化**

**Whyの有無で
効果は全然ちがう**

形だけの導入による成果と障害

• 成果

- プロセス自体の定期的改善 (振り返り)
- コミュニケーションの改善 (チーム、KANBAN)
- スコープをきることでリリースサイクルを向上 (プロダクトバックログ)

• 障害物

- スクラムのバズワード化
- POのコミット力低下
- メンバーの自発性がなかなか育たない

事例1.スクラムのバズワード化

- 「スクラム」だから〇〇やらないとダメ
- 「スクラム」をやることが目的になってしまう

原因

Whyの欠如
不安からプロセスに縛られる

対策

Whyを考えるワークショップ/聞ける環境づくり
/やってみせる

成果

Scrumではなくプロダクトを
意識できるようになった

事例2. POのコミット力低下

- 案件の投げっぱなし、情報共有不足
- POとチームの関係性悪化

原因

お互いの状況が見えていない
わからないことをすぐに聞けない

対策

まずは物理的な距離を近づける

成果

すぐに聞ける状況と、不満を口にしにくい環境

事例3.メンバーの自発性

- SMがタスクを作成、アサイン
- 振り返りがお通夜

原因

SMが自分で仕切ってしまう

対策

自己評価/他者評価のフィードバック実施

成果

自分が無意識でやっていたことへの気づき

取り組みまとめ

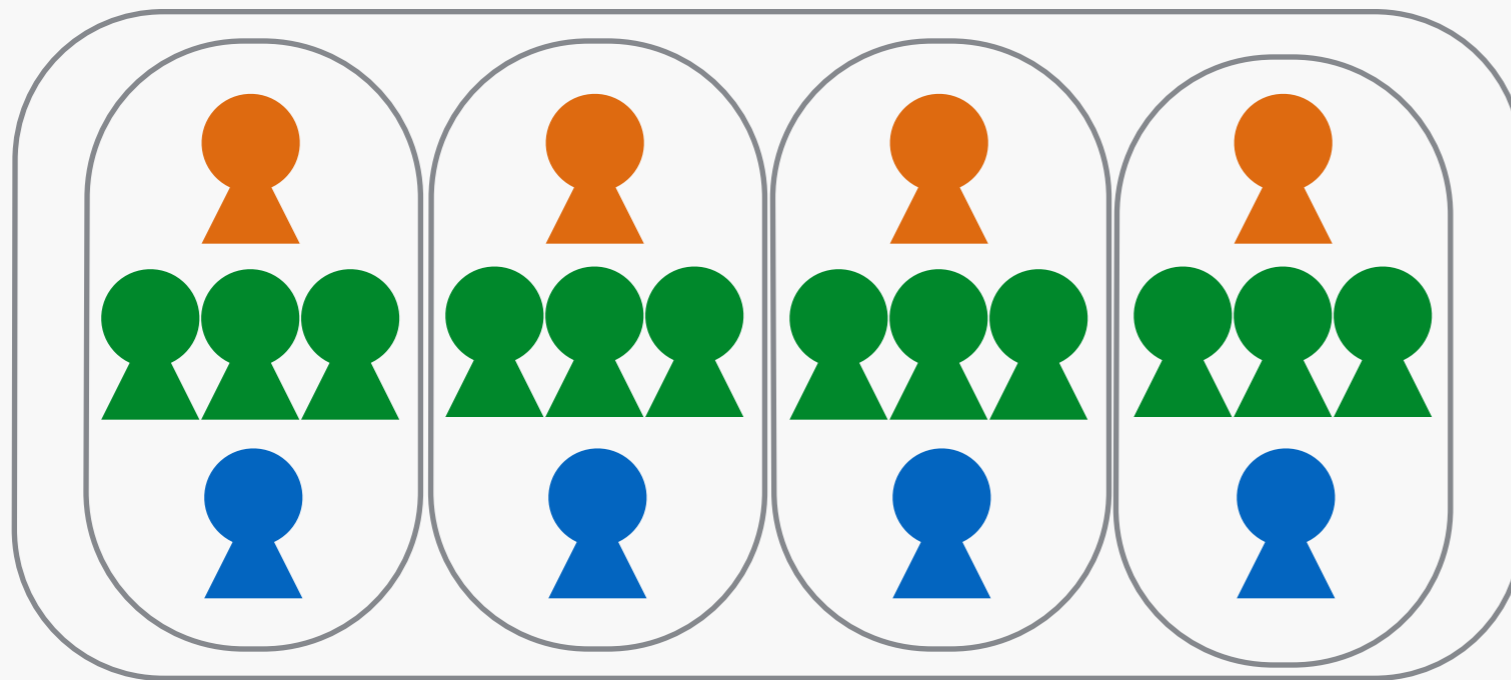
- スクラム支援T設置
 - ワークショップ、コーチング
- 席替え実施
 - POと開発チームを近くに
- 自己評価/他者評価
 - SMチェックシート実施

2.2. スクラムから大規模ア ジャイル開発へ

How we scaled

大規模アジャイル開発

全面スクラム化

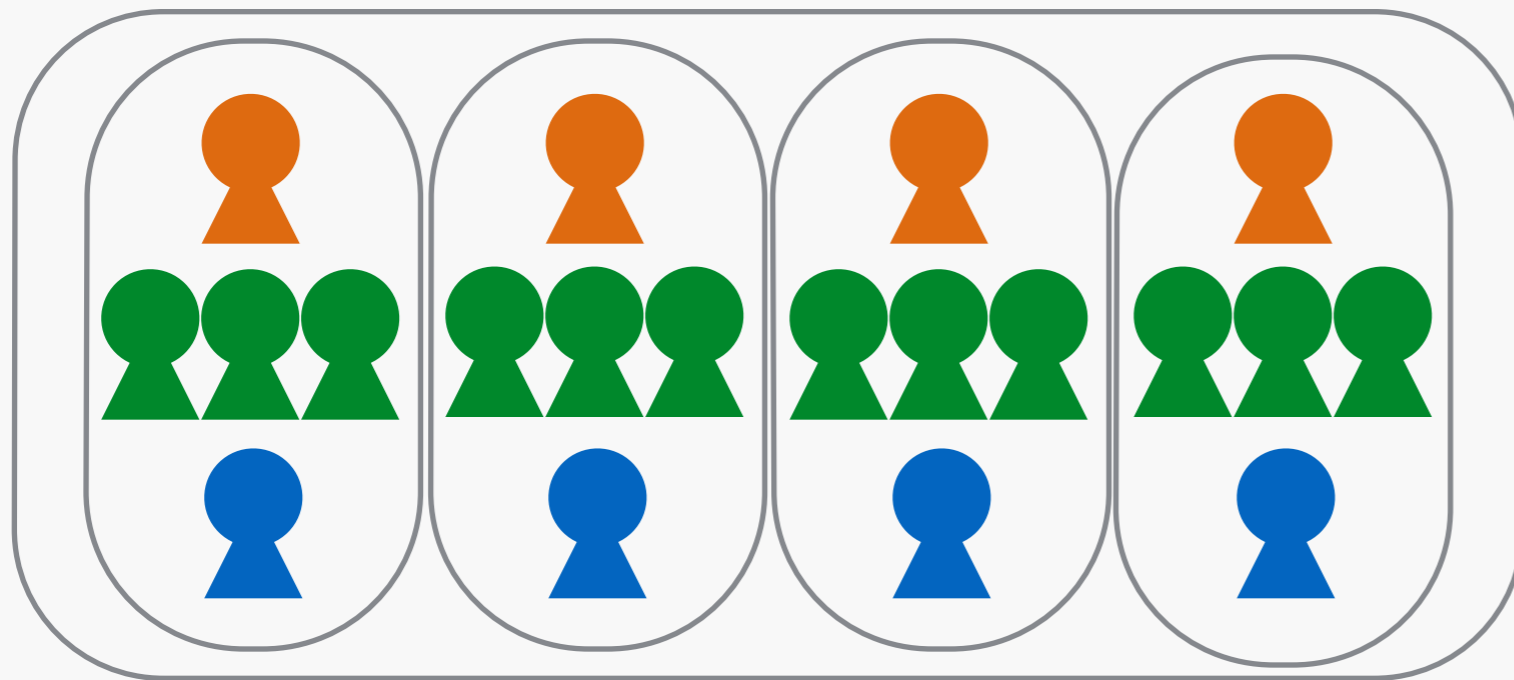


Product

POの意思を統一する人



PO Lead

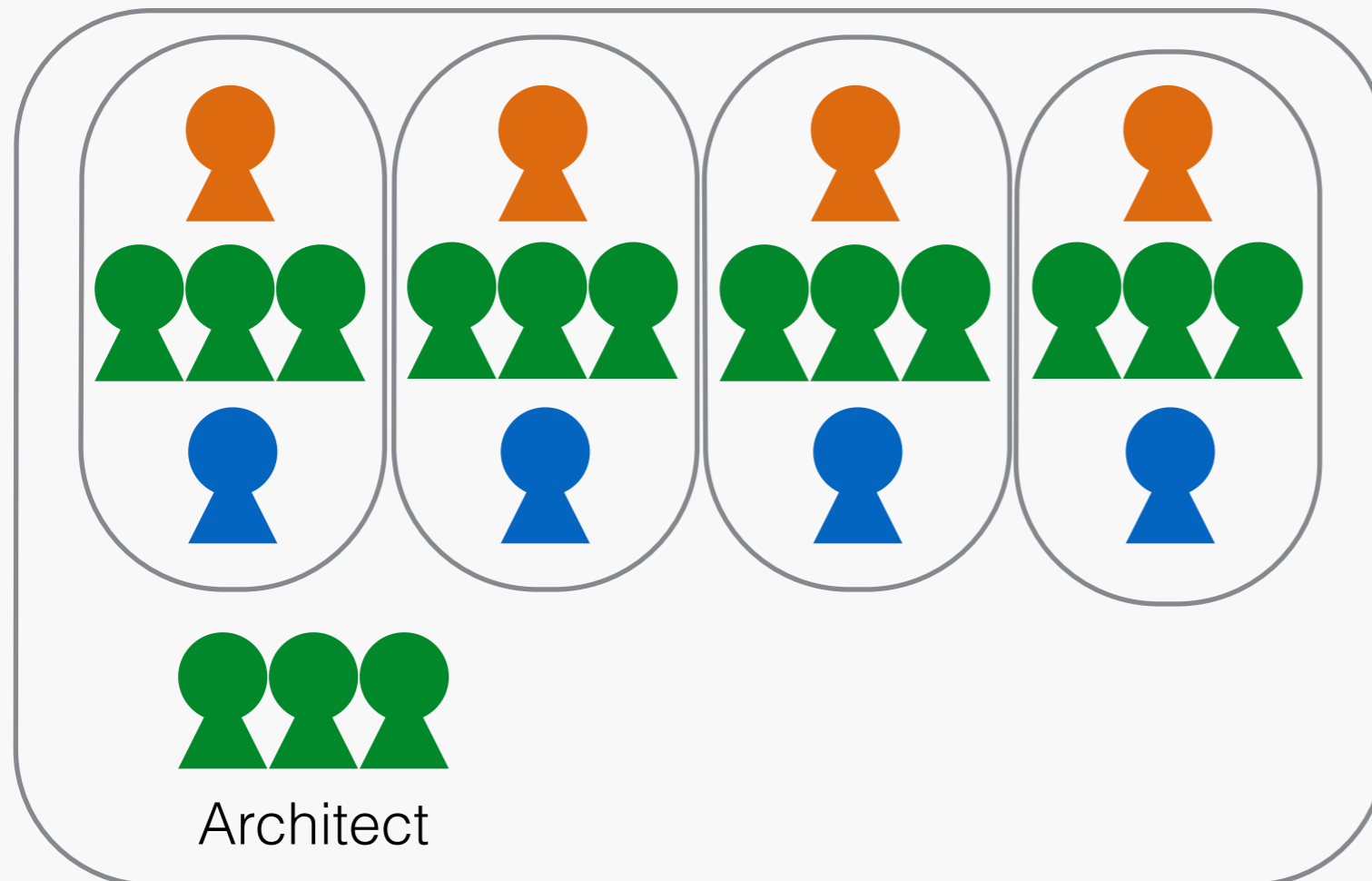


Product

アーキテクト整合性

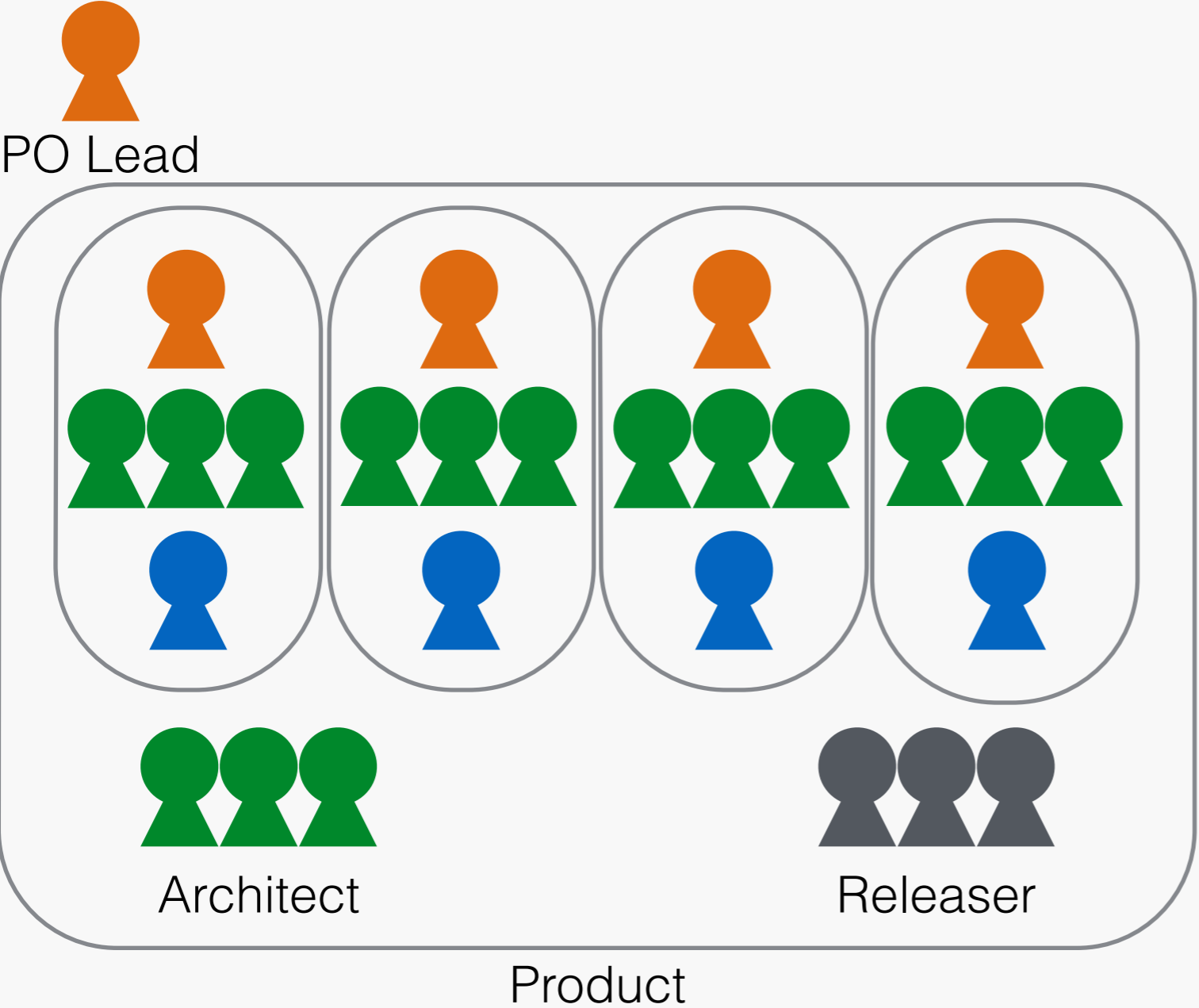


PO Lead

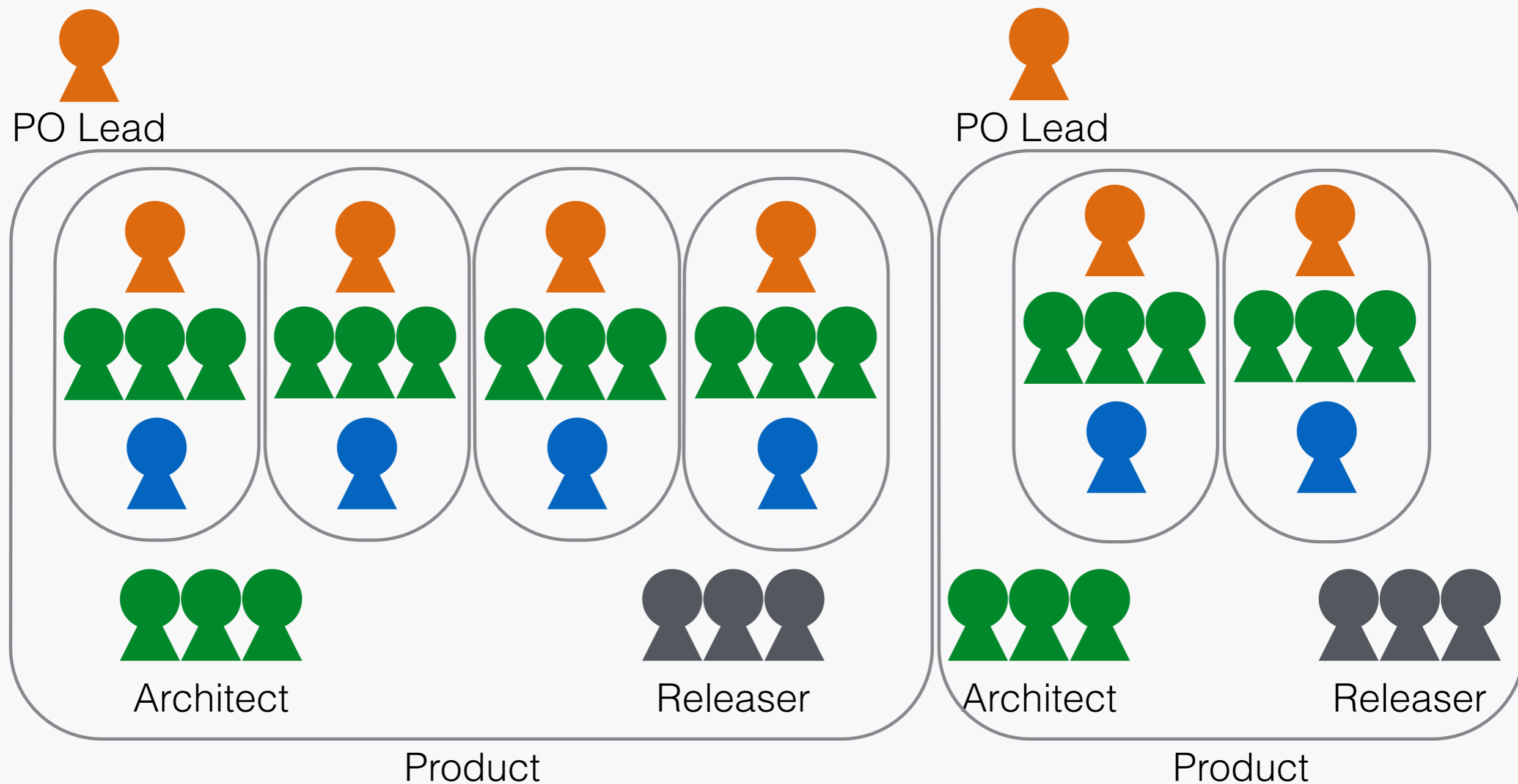


Product

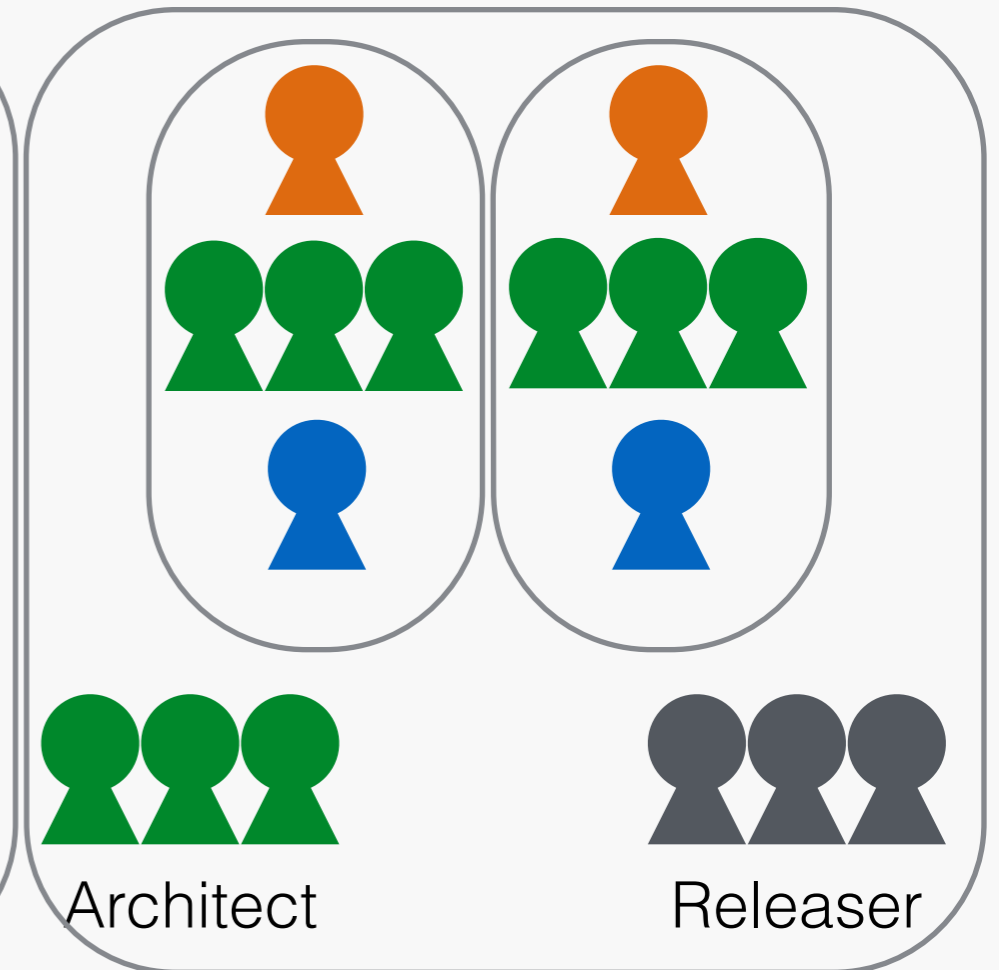
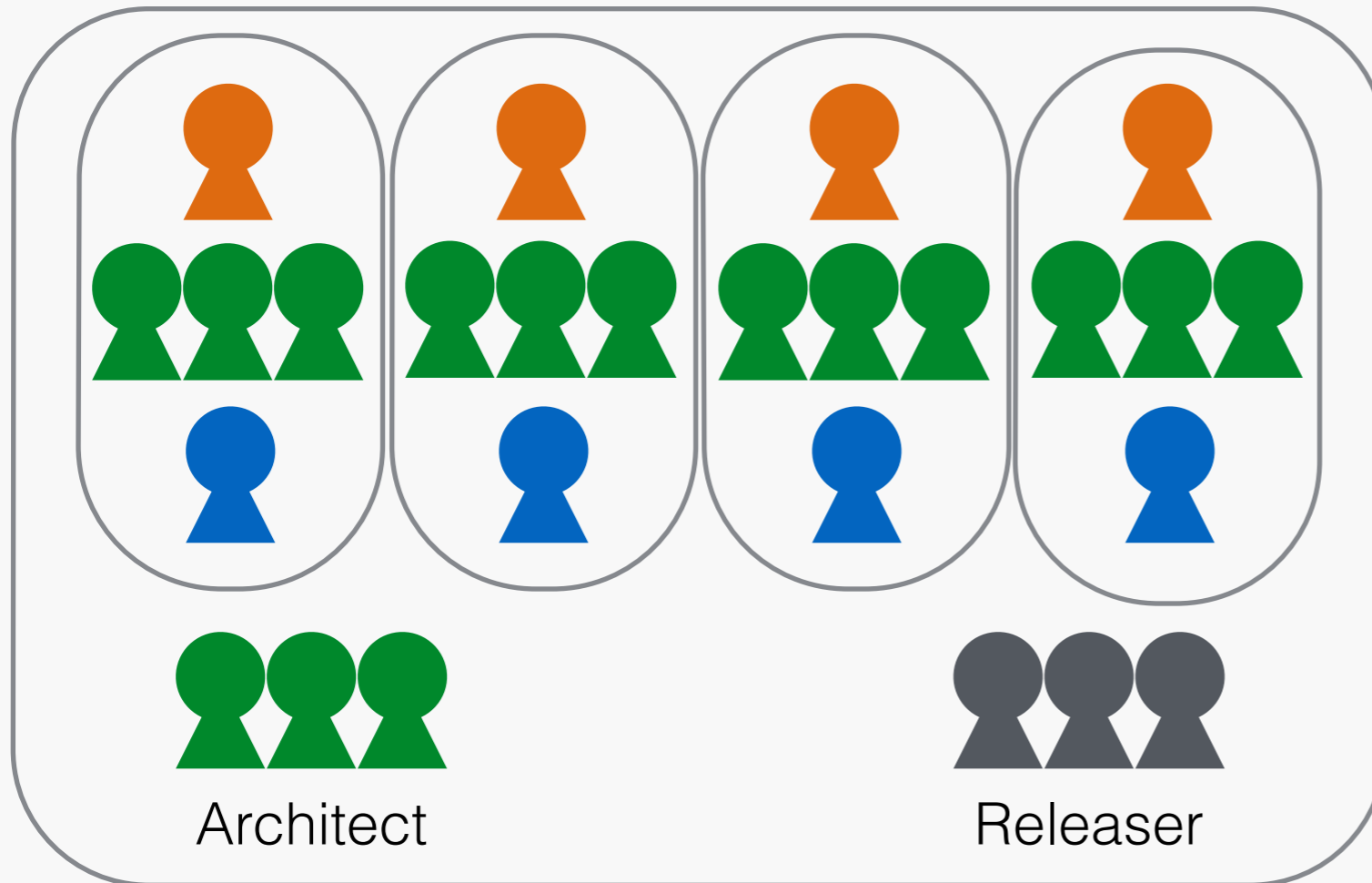
リリース案件、リリース前QA



プロダクトが増える



全体の意思を統一する人



全体のUI/UXを統一する人



Company Board



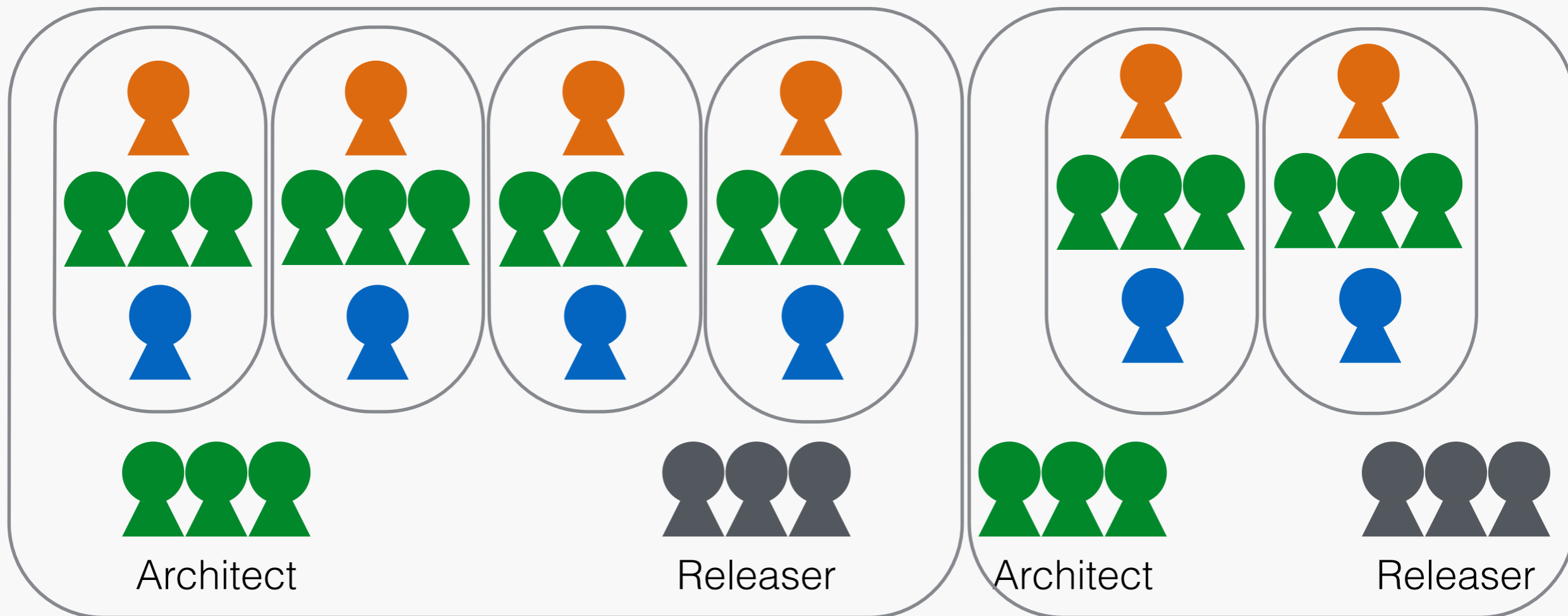
UX Specialist



PO Lead



PO Lead



Architect

Releaser

Architect

Releaser

Product

Product

全体のアーキテクト



Company Board



UX Specialist



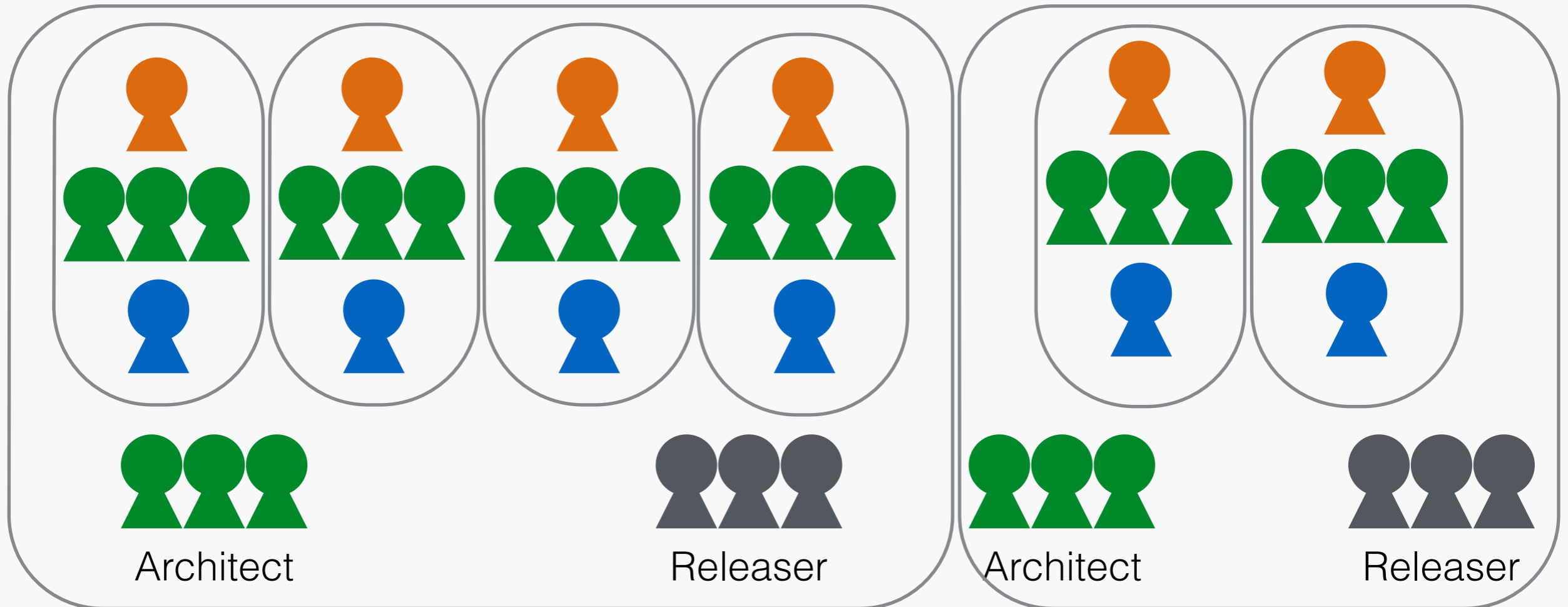
Portofolio Architect



PO Lead



PO Lead



Product

Product

全体のスクラム支援



Company Board



UX Specialist



Portfolio Architect



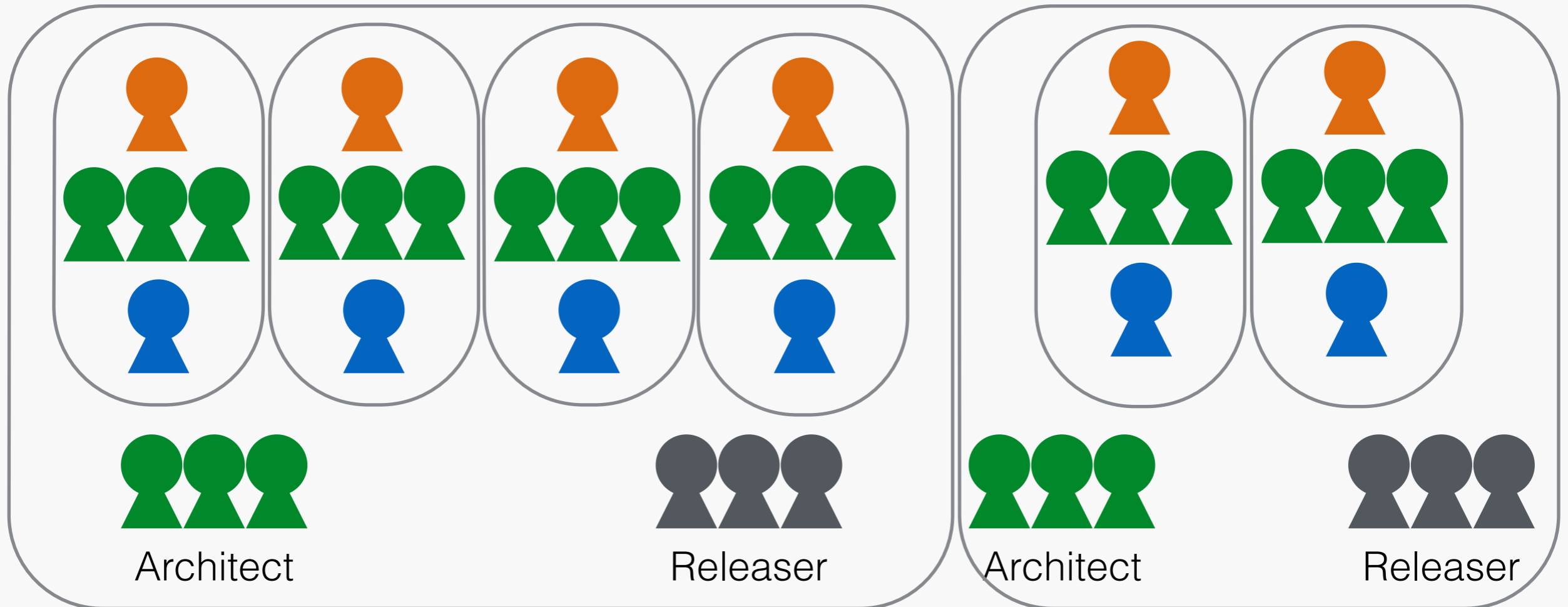
Scrum Support



PO Lead



PO Lead



Product

Product

Architect

Releaser

Architect

Releaser



Company Board



UX Specialist



Portofolio Architect



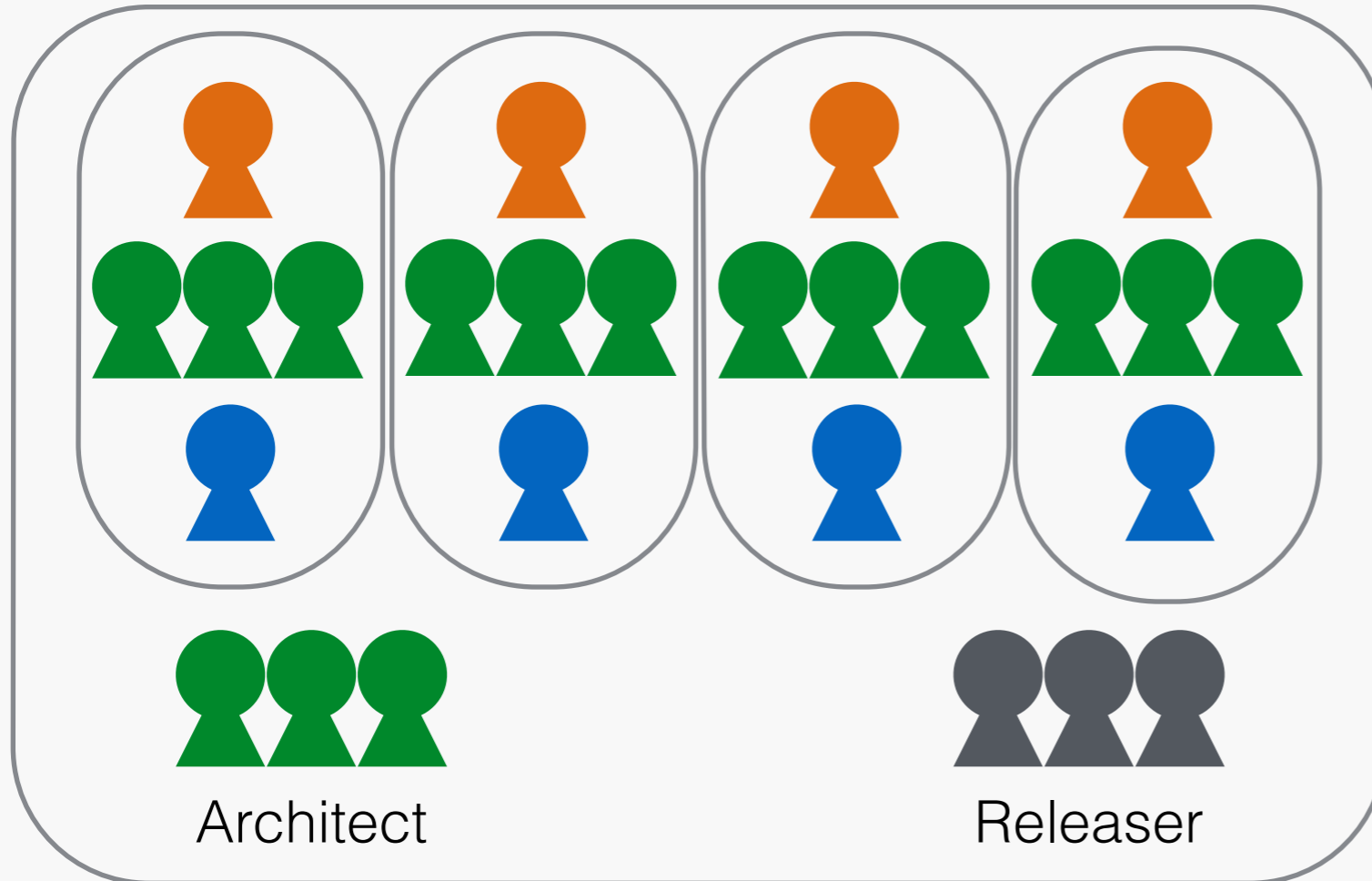
Scrum Support



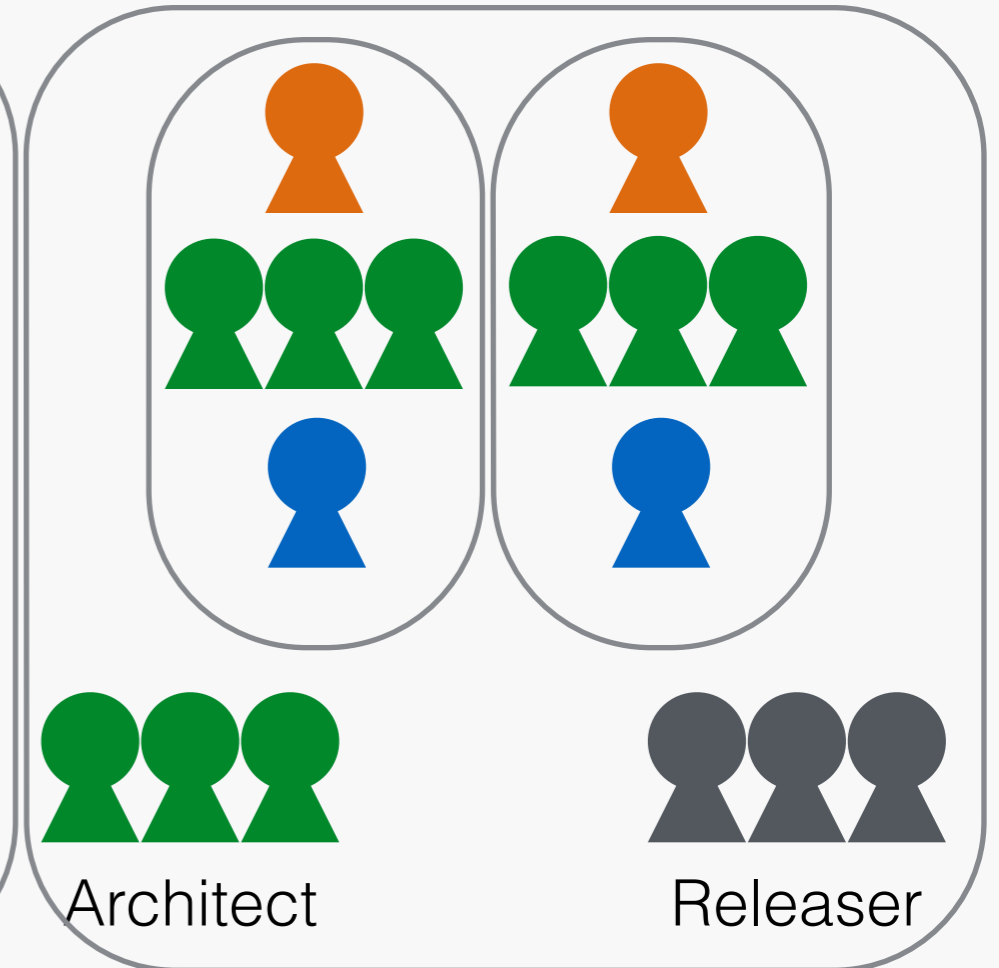
PO Lead



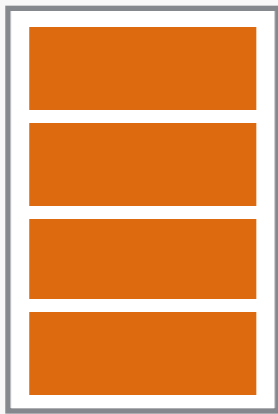
PO Lead



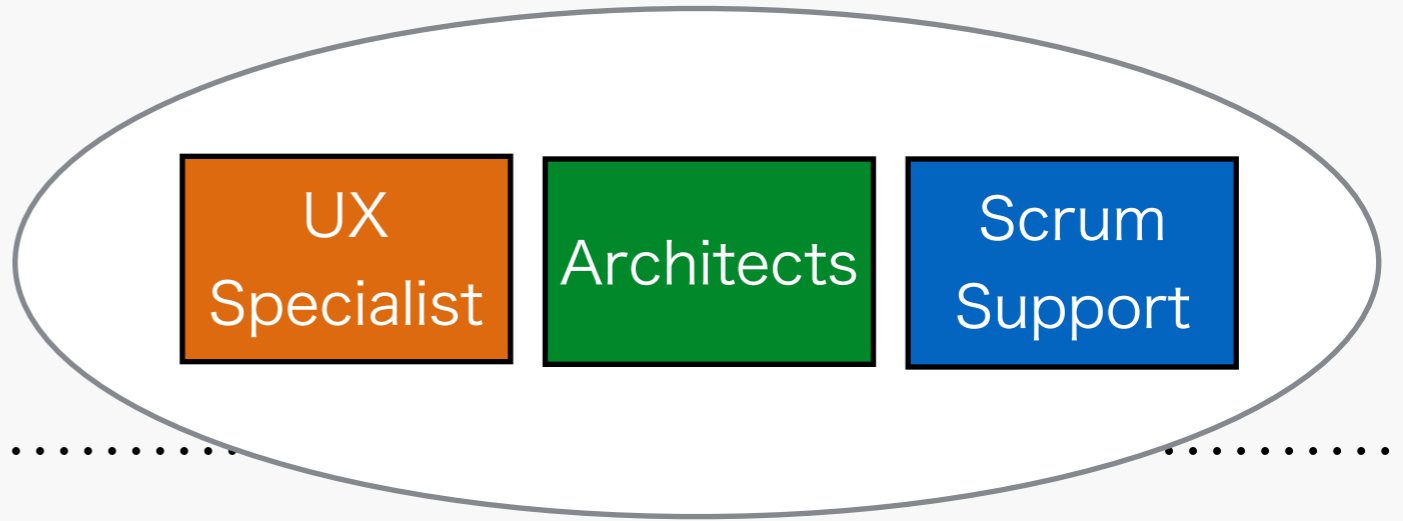
Product



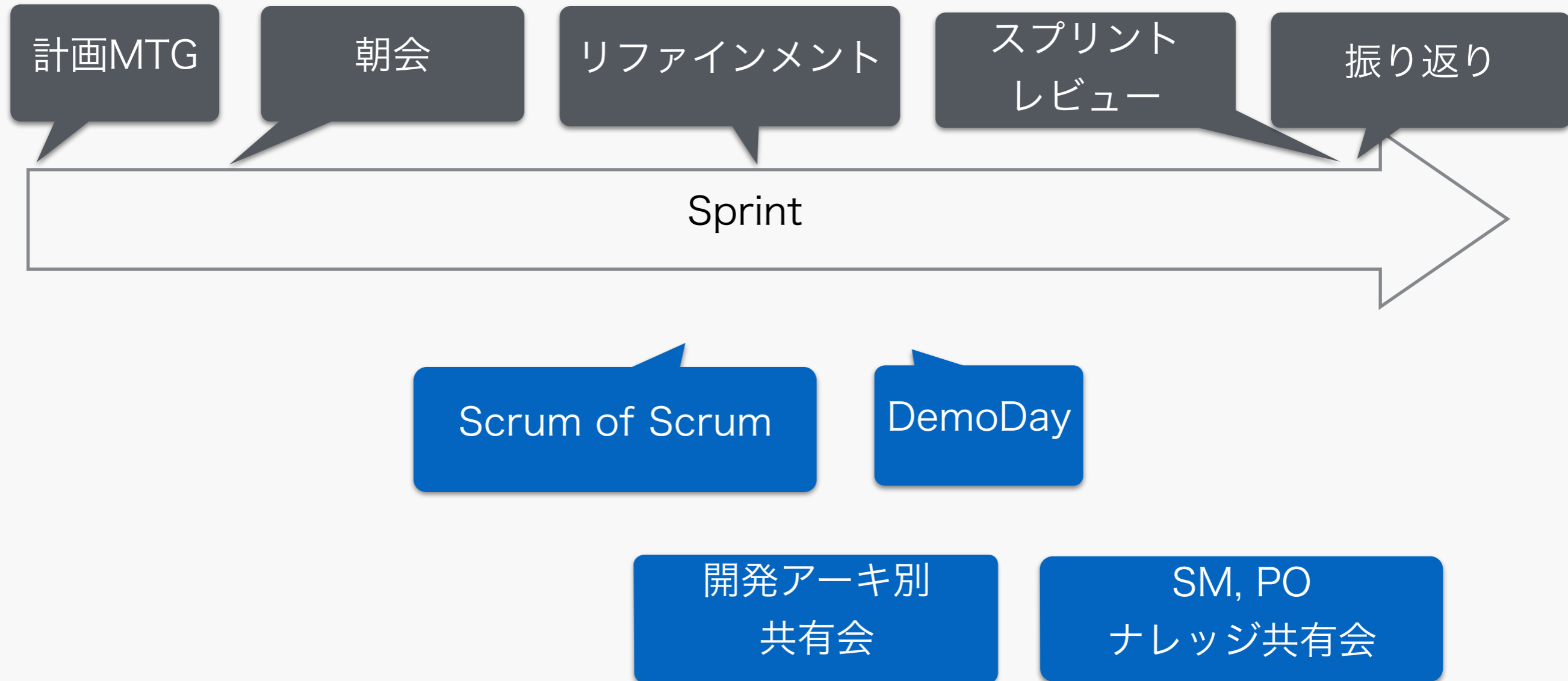
Product



Portfolio Backlog



スクラムの儀式



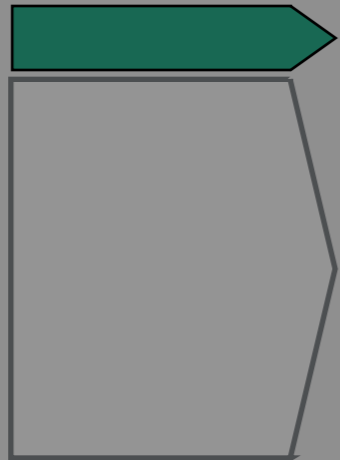
組織拡大

Product / Component

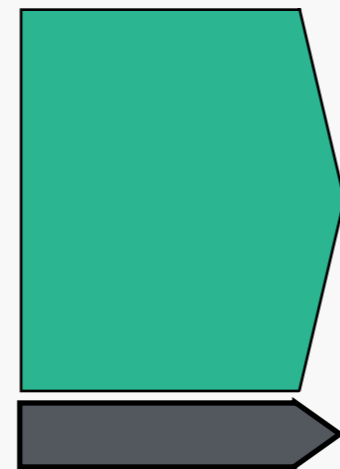
既存体制

機能組織

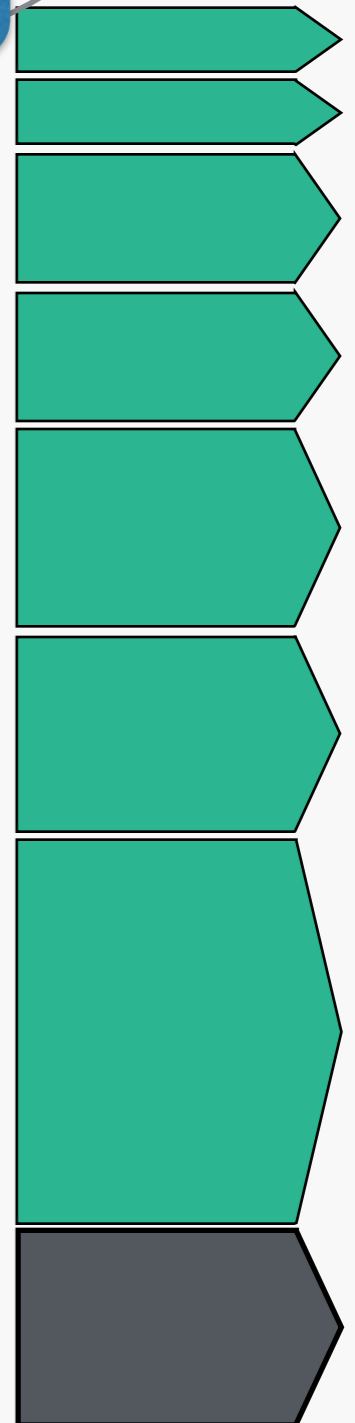
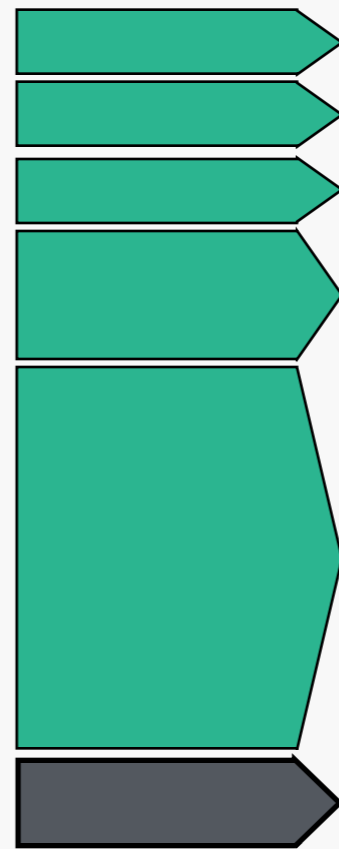
Scrum
試験導入



全面
適用



組織拡大



2013/11

2014/03

2014/06

現在

我々は何故、

- **スクラム**を行うのか？
 - 未経験の事業領域、イテレーション開発の必要性
- **スケール**を志すのか？
 - 相互に連携する複数のサービスを、全体整合性を保ちつつも並行して立ち上げたい

スクラムから大規模スクラムへ

- 成果

- 開発できる案件が増えた
- ナレッジ蓄積スピードの加速

- 障害

- 組織課題が埋もれていく
- リーダーTに対する依存
- 組織にアーキテクチャが
おいつかない

事例1. 組織課題が埋もれていく

- 全体の可視化が忘れられやすい
- 各チームだけでなく、全体最適はとれてる？

原因

チームのことにフォーカスしてしまう

対策

可視化

PO同士、SM同士で情報共有

成果

大規模アジャイル開発の改善

事例2.リーダーTに対する依存

- リーダーチームの意思決定待ち
- チーム間でのコミュニケーション不足

原因

PM：メンバー = リーダーT：チーム

対策

リーダーTの解散
※サポート型のリーダーチーム

成果

各チーム間連携の向上

事例3. 組織とアーキテクチャ

- 技術的負債の顕在化
- 人が増えてもなかなか開発速度があがらない

原因

コンウェイの法則
組織の大きさとアーキテクチャの乖離

対策

アーキテクチャの見直し
プラクティス(TDD, CI/CD)の導入

成果

挑戦中 バグ発生率は低下中

スケールアップにより発生した 問題

- チームで発生したことは抽象度があがって再発する
- 急成長するプロダクトにあわせて、組織だけでなくアーキテクチャもスケールする

4. 今後挑戦したい課題

Our current impediment list

事例1. 組織課題が埋もれていく

- 全体の可視化が忘れられやすい
- 各チームだけでなく、全体最適はとれてる？

原因

チームのことにフォーカスしてしまう

対策

可視化
PO同士、SM同士で情報共有

成果

大規模アジャイル開発の改善

**改めて、
検証と適応と透明性**

組織拡大と可視化の重要性

- 組織拡大 = 意思を持つ人の増加
- 可視化は「事実」を顕在化させる
 - 意見は受け入れづらいが、事実は受け入れやすい
 - 目的が共有できていれば、課題を解決する意思を共有できる

改めて、検証と適応と透明性

1. プロセスの透明性

2. 成果の透明性

1. プロセスの透明性

- 大規模化に伴って登場人物が増えたが、アジャイルさの阻害要因になっていないか？
- 個別のレビューも塵が積もれば山となる

全体のスクラム支援



Company Board



UX Specialist



Portofolio Architect



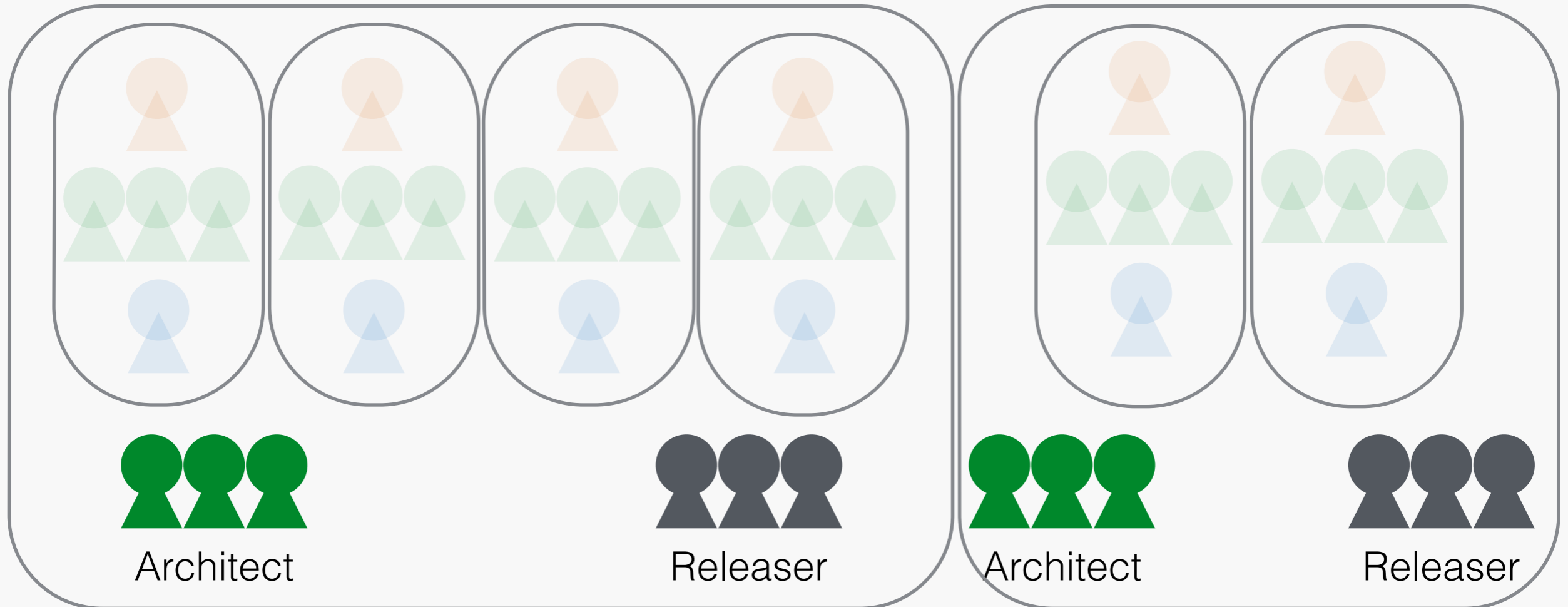
Scrum Support



PO Lead



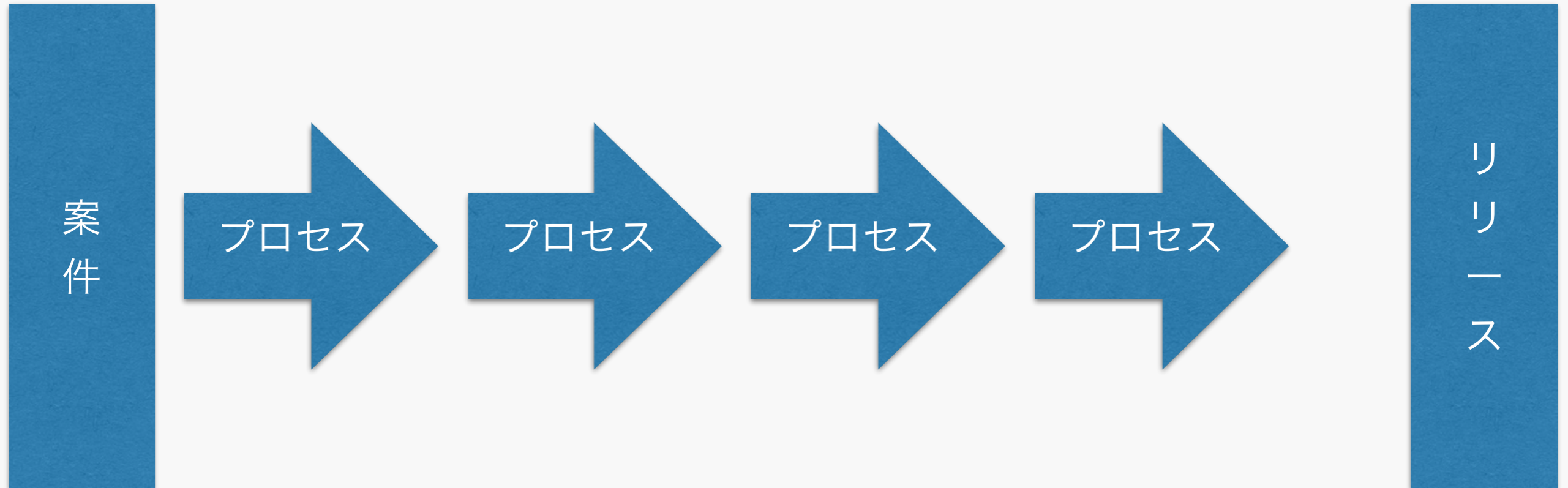
PO Lead



Product

Product

大規模のプロセスを どう可視化するのか？



- どれだけのプロセスを実践している？
- 各プロセスにどれだけコストがかかっているのか？
- そのプロセスの意味と目的は？権限移譲出来ない？

2.成果の透明性

- 実際に4つのプロダクトを並行開発しているが、本当にROIは最大化できているか？
- 並行案件数が増えるほど、成果の切り分けは難しくなってしまう

大規模の成果を

どう可視化するのか？

- 改めて案件毎のKPI寄与を個別に可視化
- 全体としてのKPI向上のために、より精度の高い戦略を立てるための定量的な素材を用意する

改めて、検証と適応と透明性

1. プロセスの透明性

2. 成果の透明性

5. まとめ

Summary

スクラムから学んだこと

- Whyを理解/共有をすること
- 物理的な距離を近づけること
- PMをそのままSMにしないこと (SMは我慢すること)

大規模化から学んだこと

- 全体の可視化を改めて行うこと
- リーダーチームもサポート型に
- 組織構造にあわせたアーキテクトを作ること

まとめ

スケールアップしても

スクラムの原理原則はかわらない

スクラムの導入に最も重要なのは「Why」

検証と適応と透明性

失敗から学んだことを
お伝えしてきましたが

結局やってよかった？

YES!

Scale Upしてよかったこと

- 仲間が増える
- できることが増える
- 学ぶことが増える

出来る、大規模アジャイル開発

- 検証と適応と透明性
 - 失敗から学ぶことが出来れば大丈夫！

出来る、大規模アジャイル開発

- 検証と適応と透明性
 - 失敗から学ぶことが出来れば大丈夫！

一つでも皆様のお役に立てるものがあれば
幸いです

課題を解決する仲間 絶賛募集中！

- <http://www.career.recruit-lifestyle.co.jp/>