

メンバーの成長を促進する 組織マネジメント

三菱電機株式会社 細谷 泰夫

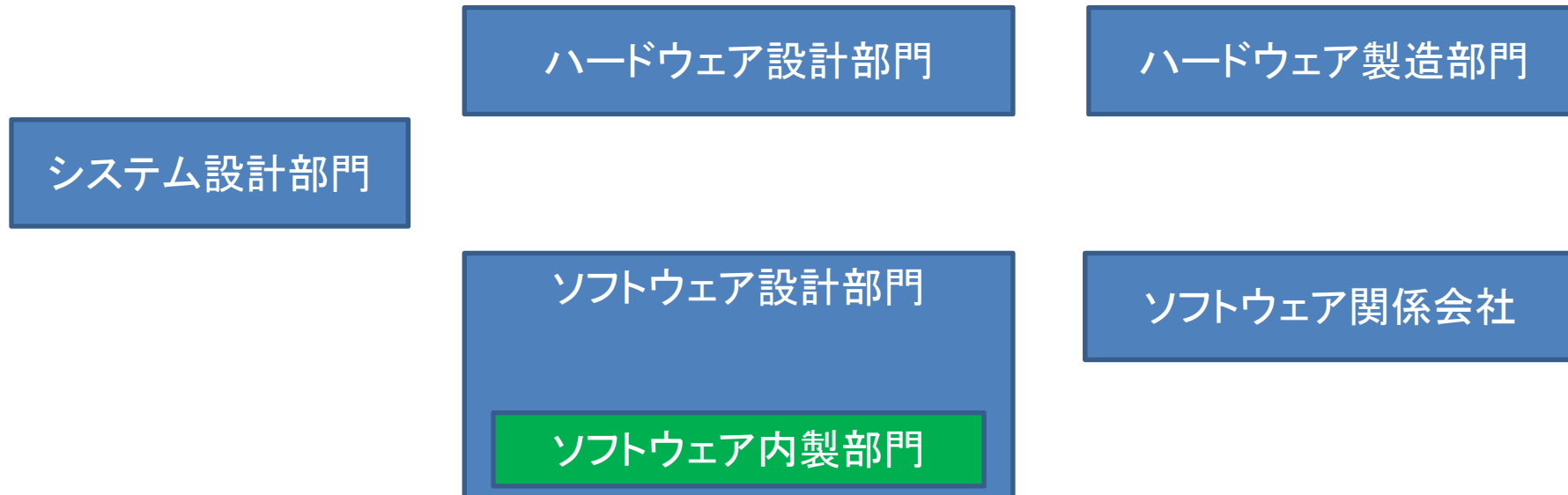
三菱電機株式会社

- 複数の機器で構成するシステムの開発
- ソフトウェアは、独自に開発するハードウェア、汎用のボードコンピュータ、普通のPC (Linux/Windows®) など機器によって様々。



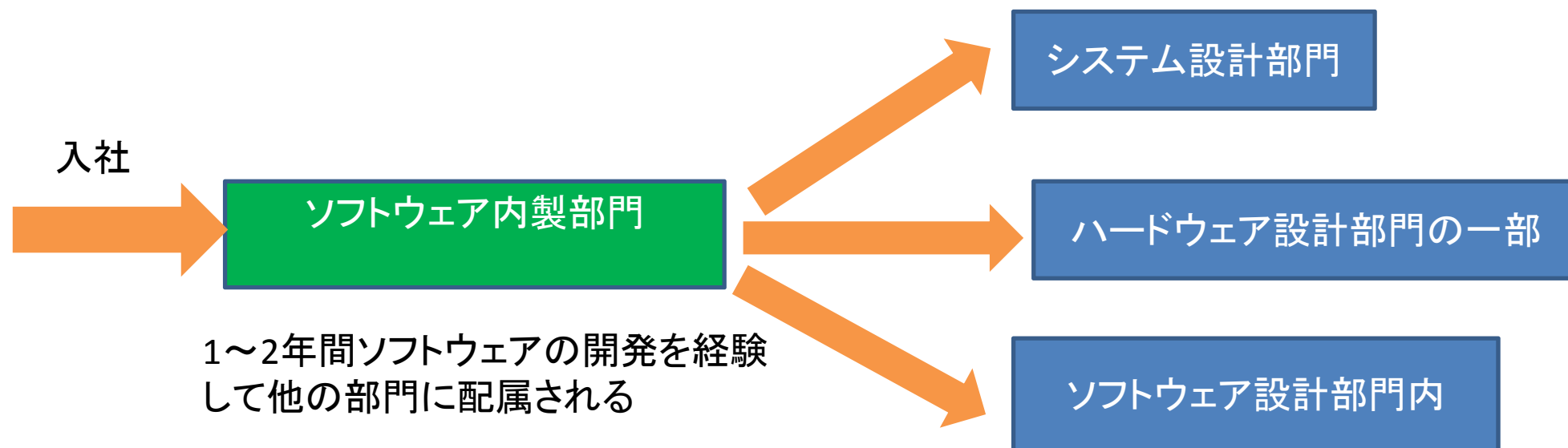
組織の構成

- 各プロジェクトのマネジメント、システム設計を担当する部門
- ハードウェア設計部門
- ソフトウェア設計部門、ソフトウェア関係会社



私がマネジメントしている組織

- システム設計部門、ハードウェア設計部門の一部、ソフトウェア設計部門の新人が1～2年が配属される。

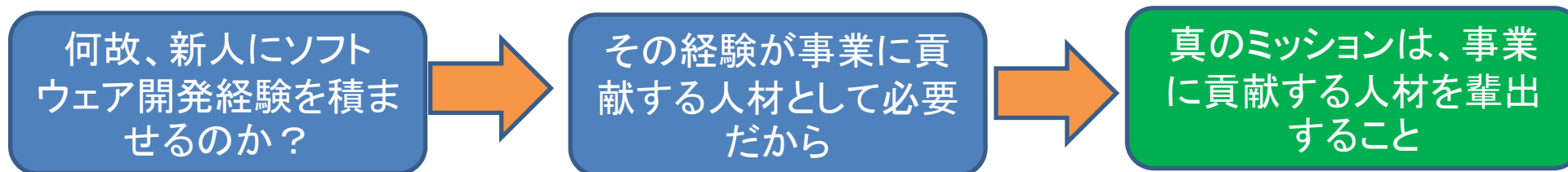


組織のミッション

- 新人の育成(ソフトウェアの開発経験)
- ソフトウェアの内製を通じた事業への貢献

組織のミッション

- 私なりの組織のミッションの解釈



しかし、1～2年でそのような人材になるのは難しい。



継続的に成長する必要がある。
成長するのは本人次第。だが、最初の1歩を踏み出す支援はできる

個々のメンバーの成長のエンジンを点火する

組織のミッション

- 成長するには、良い仕事をすることが必用。
- 良い仕事とは、自分たちのアイデアを生かして、成果を得る経験が積めて、達成したことの価値が評価される仕事。

高い目標

目標とする組織の姿

新人中心のメンバーが、自分たちのアイデアを生かして、仕事で価値ある成果を出すことによって成長を実感し、さらなる成長に向けて自ら歩みはじめることができる

これは、簡単なことではない

経験が浅いメンバーに対して一般的に抱く不安

- ・コストが高くなる？
- ・品質は大丈夫？

このイメージを克服するためにあえて高い目標を設定

ありたい姿になるための高い目標

(通常の組織と比較して)2倍以上の効率と、圧倒的な品質を実現する。

メンバーの特徴

- 新人中心のメンバー

弱み：経験が浅い

強み：伸びしろが大きい。

モチベーションが高い。

固定観念がない。



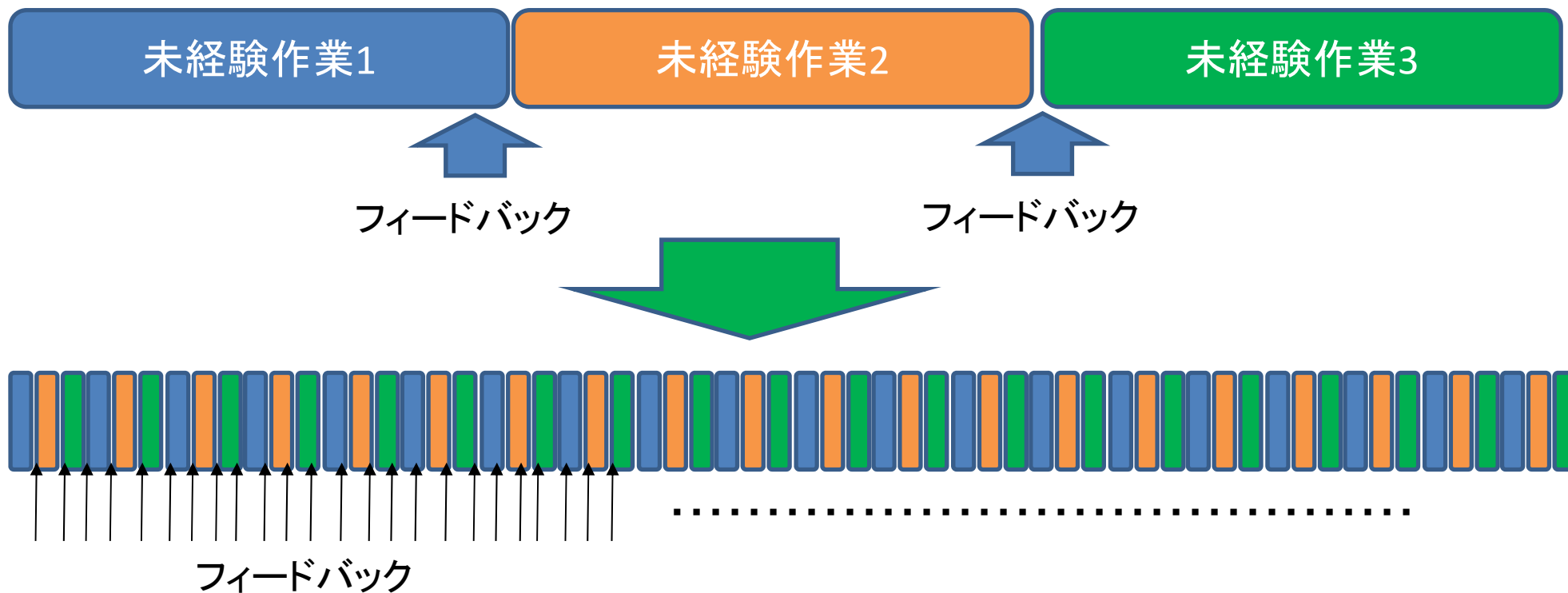
開発が成功するには、短期間での成長が必須。

あるプロジェクトで成長してそれを次に活かすのでは遅い。

最初のプロジェクトの実行中に大きく成長し、同時にそのプロジェクトの成功に貢献してもらう。

どうやったら成長するか？

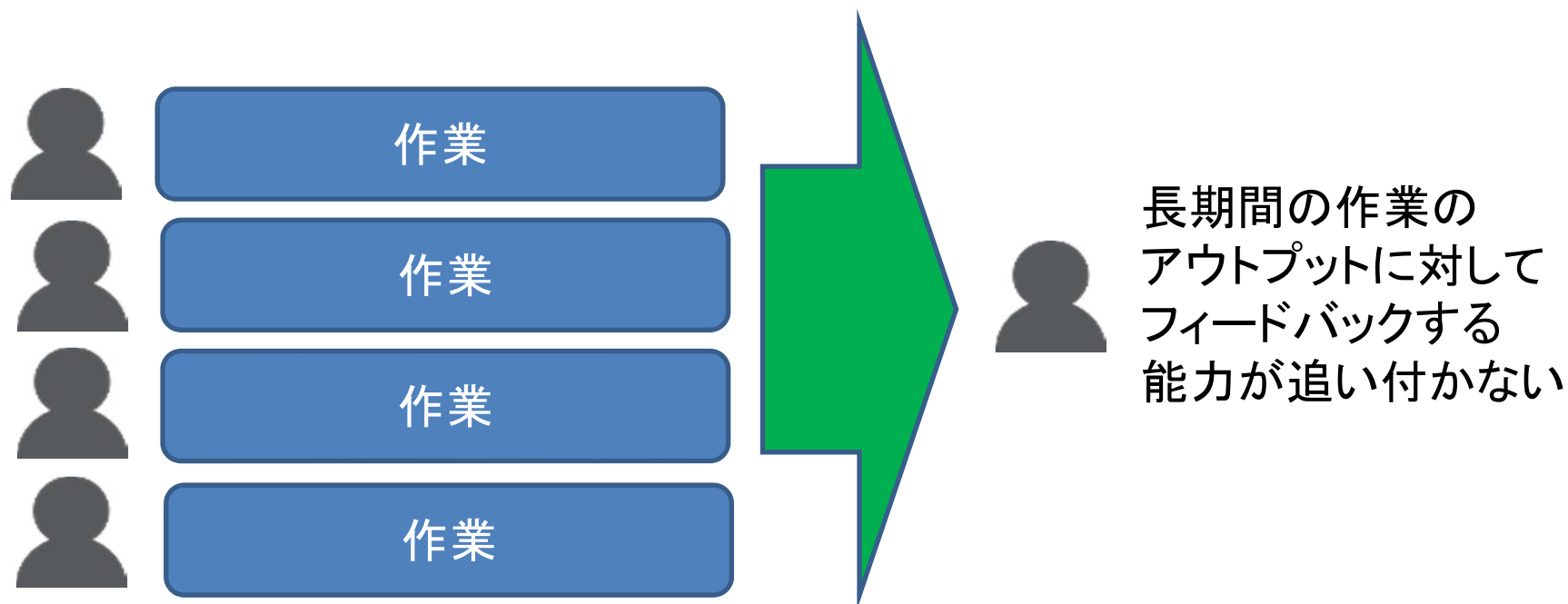
- 短期間での繰り返しと頻繁なフィードバック



一連のプロセスを繰り返し実行することによる学習効果。
頻繁にフィードバックすることによる教育効果

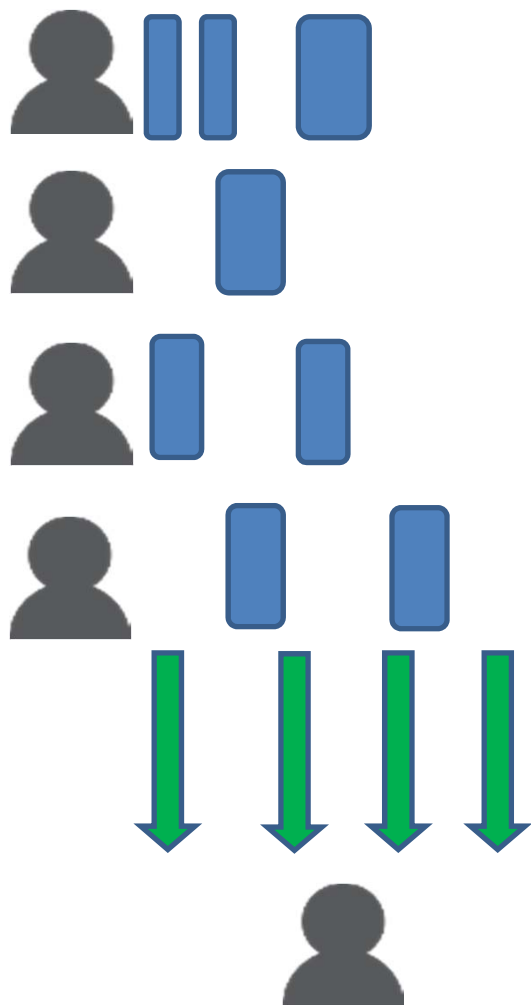
フィードバック中心のフロー制御

- 20人中15人が1～2年目の組織
- 経験者からのフィードバックが重要



経験者がボトルネックになりやすい。

フィードバック中心のフロー制御



一度の作業期間を短くすることで
経験者のフィードバック能力の許容範囲を
保つことができる

新人の成長を促進する手段としてアジャイル
開発が有効

技術的なチャレンジ

人手でやっていることを自動化することによる大幅な効率化と品質向上

- メタプログラミング
- テスト自動化
- サーバー構成自動化

組織の方針を個人目標に反映する



プロジェクトは各組織の
メンバーにより構成されてい
る



個人の目標はプロジェクトの
QCDのみとなりがちだった。

組織の方針を個人目標に反映する

- 担当のプロジェクトの成功に貢献することはもちろん大切だが、それだけでは不十分。



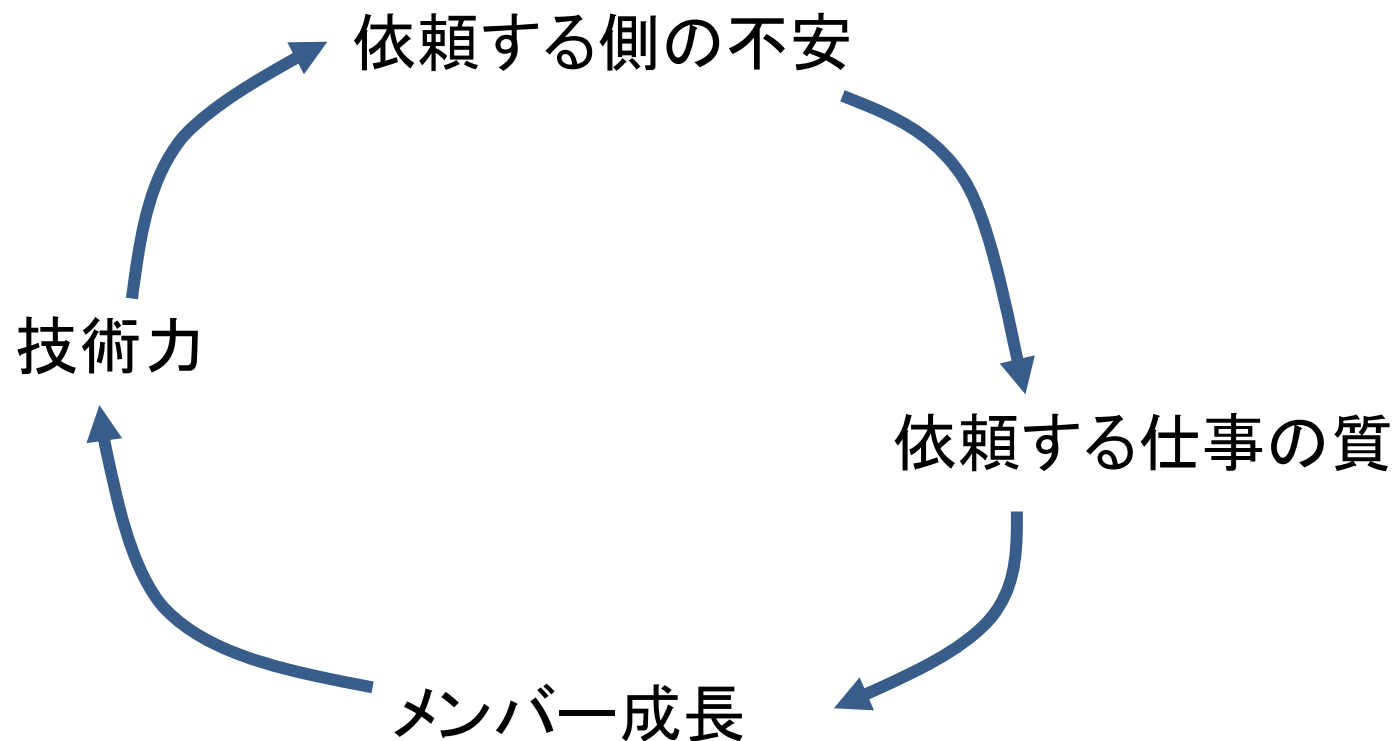
組織のビジョンを共有して、個人目標に反映して行動を促していくための方法が必要。

組織の方針を個人目標に反映する

- 組織のメンバー全員が参加するワークショップを開催。
- 組織の置かれている背景をループ図を用いて表現して共有。
- ループ図を正回転させるために、どの部分に力を加えるかをメンバーで話し合う。
- 話し合った内容を元に「組織として今年1年でなりたい状態」を4人程度のグループ毎に決める。
- 「Fearless Journey」というワークショップで「なりたい状態」に対する障害とその対処を共有する。
- ワorkshopで感じたことを元にした目標を最低1つは、個人の目標に加えてもらう。

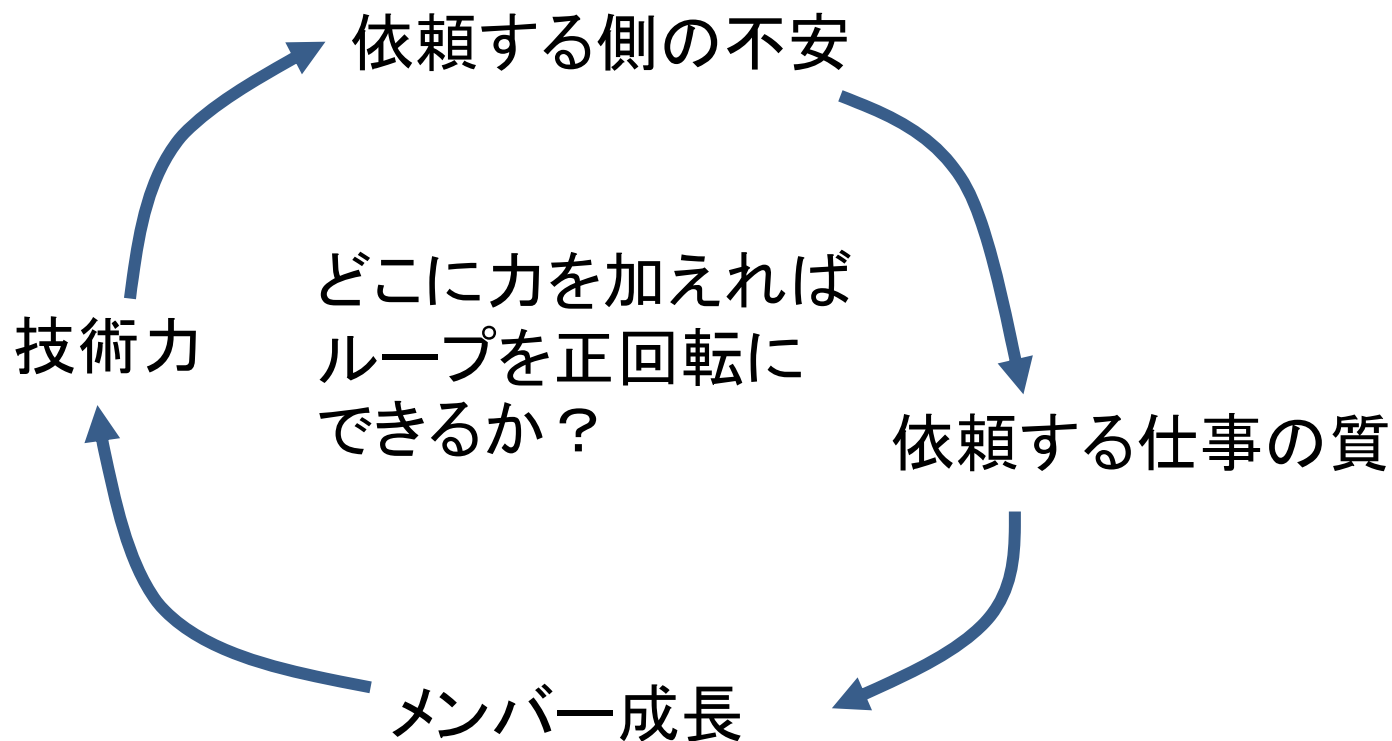
(1) ループ図で組織の状況を共通する

例)



ワークショップ

(2) ループ図のどこに力を加えれば良いかを話し合う。



(3)「Fearless Journey」を実行する。ただし、チームのゴールは、(2)の議論を踏まえて決める。

Fearless Journeyの戦略カードと手順は以下を参照。

<http://waicrew.thebase.in/items/37055>

(4) 個人目標に最低1つ、ワークショップを踏まえた目標を考えて設定してもらおう。

チャレンジできるチーム作り

- 新しいやり方で今できていないことをできるようにするのは価値が高い
- 新たな試みをどんどんと実施して、仕事に活かしていく。⇒成長に繋がる



実験的アプローチによる改善活動により新しい手法にチャレンジする。

新手法をチームに取り入れて改善するやり方

- JaSST '13 Kansai

<http://jasst.jp/symposium/jasst13kansai/report.html>

- JaSST'13 Hokkaido

<http://jasst.jp/symposium/jasst13hokkaido/report.html>

- 日科技連 「ソフトウェア品質のホンネ」

http://www.juse-sqip.jp/wp3/honne/backnumber_162/

改善 = 変化

改善するときにはやること。

- 「やることを増やす」
- 「今やっていることを変える」
- 「今やっていることをやめる」



これらは、現状に対する「変化」と言える。

変化への恐れ

- ほとんどの場合、「現状」には問題はあるが、“なんとかなっている”
- 「変化」は、現状なんとかなっているバランスを崩す可能性がある。



それ故に、「変化への恐れ」が生まれる。

特に、組織にとって、新しい手法を適用する場合にこの「恐れ」が大きくなる。

改善のためには、この「恐れ」を乗り越えなければならない。

「恐れ」への典型的な対処

- 組織的な合意のプロセスを踏むことで「恐れ」を克服する。
- 事前に入念な調査を行い、費用対効果を組織で合意し、パイロットプロジェクトを選定して手法を評価する。

典型的なプロセスの問題点

- 適用までの投資が大きい。
情報収集の開始から実プロジェクトへの適用までにかかる時間とコストが大きい。
- 撤退しにくい。
適用までの投資が大きいので、(特に提案者は)撤退の選択をしづらくなる。

新手法適用のジレンマ

- 自分たちのコンテキストに合わせた形で手法を適用しなければ効果はできにくい
- 自分たちのコンテキストでの実践をしなければ、手法を自分たちのコンテキストに合わせてどのように適用すればいいかはわからない。



最初のパイロットで、未経験の手法を実践しながら自分たちのコンテキストに合うように適用して、良い結果を出さなければならない。

実験的アプローチは、 新手法適用時のジレンマを 解決する手段

実験的アプローチの三原則

実験的アプローチによる現場改善において
重視する3原則

原則1: 「改善の範囲をきわめて小さくする」

原則2: 「実践の結果を素早く得る」

原則3: 「結果を評価して、軌道修正する」

原則1 改善の範囲をきわめて小さくする

- 改善の範囲が大きいと、改善にともなう変化量、手段の複雑度が大きくなる。
- 変化量、複雑度が大きいと、結果を得るまでのコスト、時間が大きくなる。



改善の範囲を極力小さくすることで、改善による変化量、手段の複雑度を小さくする。

原則2 実践の結果を素早く得る

- 原則1「改善の範囲をきわめて小さくする」を適用することにより実践の結果を素早く得ることができる。



結果を素早く得ることにより、素早く軌道修正することができる。

原則3 結果を評価して軌道修正する

- 原則2「実践の結果を素早く得る」より早期に得られた結果を評価する。
- うまくいっていれば継続、改善点があれば変更、効果がわりにあわないと判断すれば取りやめを判断する。



- ・素早くやめることもできるため、新手法適用のリスクを低減できる。
- ・取りやめた場合でも、置かれているコンテキストとその課題と手法の関係性を実践結果から学習することができる。

- 情報収集・調査
- 課題の発見
- 課題と手段のマッチング
- 手段の試行
- 手段の実践
- 結果の評価

情報収集・調査

- “良さそうな手法”を発見する活動
- ターゲットを絞らない「収集」とターゲットを絞った「調査」がある。
- 「収集」では、自分に役に立つかどうかは気にしない
- 「収集」の情報源は、カンファレンス、書籍、SNS。
誰からどんなキーワードが出ているかをチェック。
(AさんとBさんが“XXXX”がすごい！と言っている
など。)
- 「収集」は定常的に行う。
- 「調査」は、課題と手段のマッチングによって特定の
対象が絞られている場合に実施する。

課題の発見

- 反復的な活動を利用する。
- アジャイル開発のイテレーションとの親和性は高い。
アジャイル開発でなくても作業に繰り返しの要素を入れると効果的。
- 繰り返しの中で未来の反復でも繰り返し現れるような課題を発見する。具体的かつスコープが小さい課題。

課題と手段のマッチング

- 発見した課題と手段をマッチングさせる。
- 情報収集により発見した手法が手段になる場合と発見した課題のスコープに合った手段をここで抽出する場合がある。
- 課題に合う手段は一つではないし、手段が有効な課題も一つではない。

手段の試行

- 課題とマッチングした手段が対象。
- マッチングにより、適用のスコープがより絞られていることが重要。
- 課題を解決する手段を一つではないので、手段を試行するのと並行して別の手段を実践する場合もある。
- 試行のゴールは実践を開始できる状態になること。試行しなくても実践を開始できるものはこの活動を省略する。

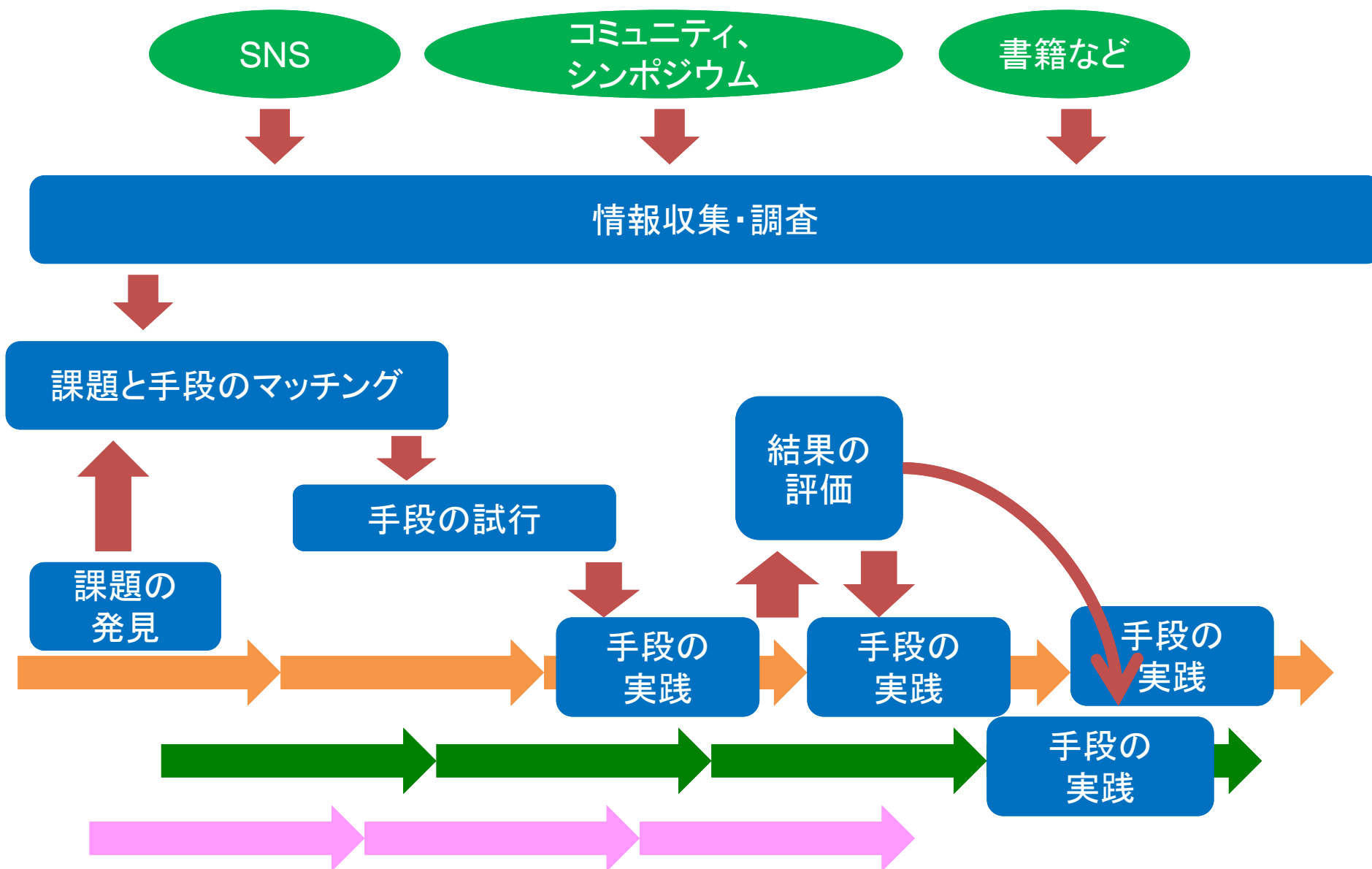
手段の実践

- 実際の業務の中で課題にスコープを絞った手段を実践する。
- 実践による結果が極力素早く得られるようにすることが重要。

結果の評価

- 実践により得られた結果を評価する。
- 反復のふりかえりの際にその効果や改善点について話し合うのが効果的。
- 当初は考えていなかった問題や、意識していなかった副次的な効果を発見。(実践してこそ得られる)
- 結果を評価して、手段を継続するのか、軌道修正するのか、やめるのかを判断する。
- ここでの気づきからより手段を発展させるアイデアが出る場合もある。

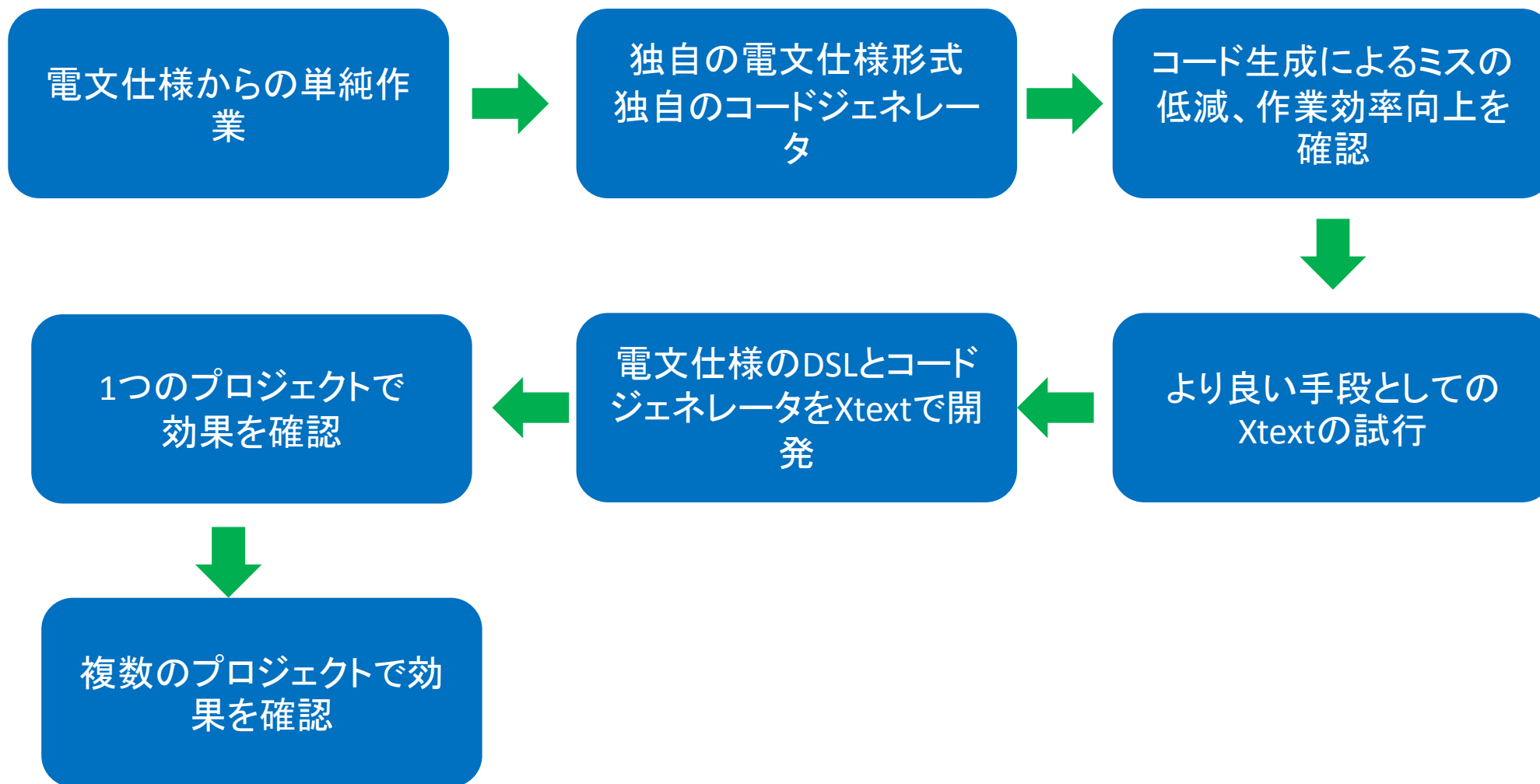
活動の全体像



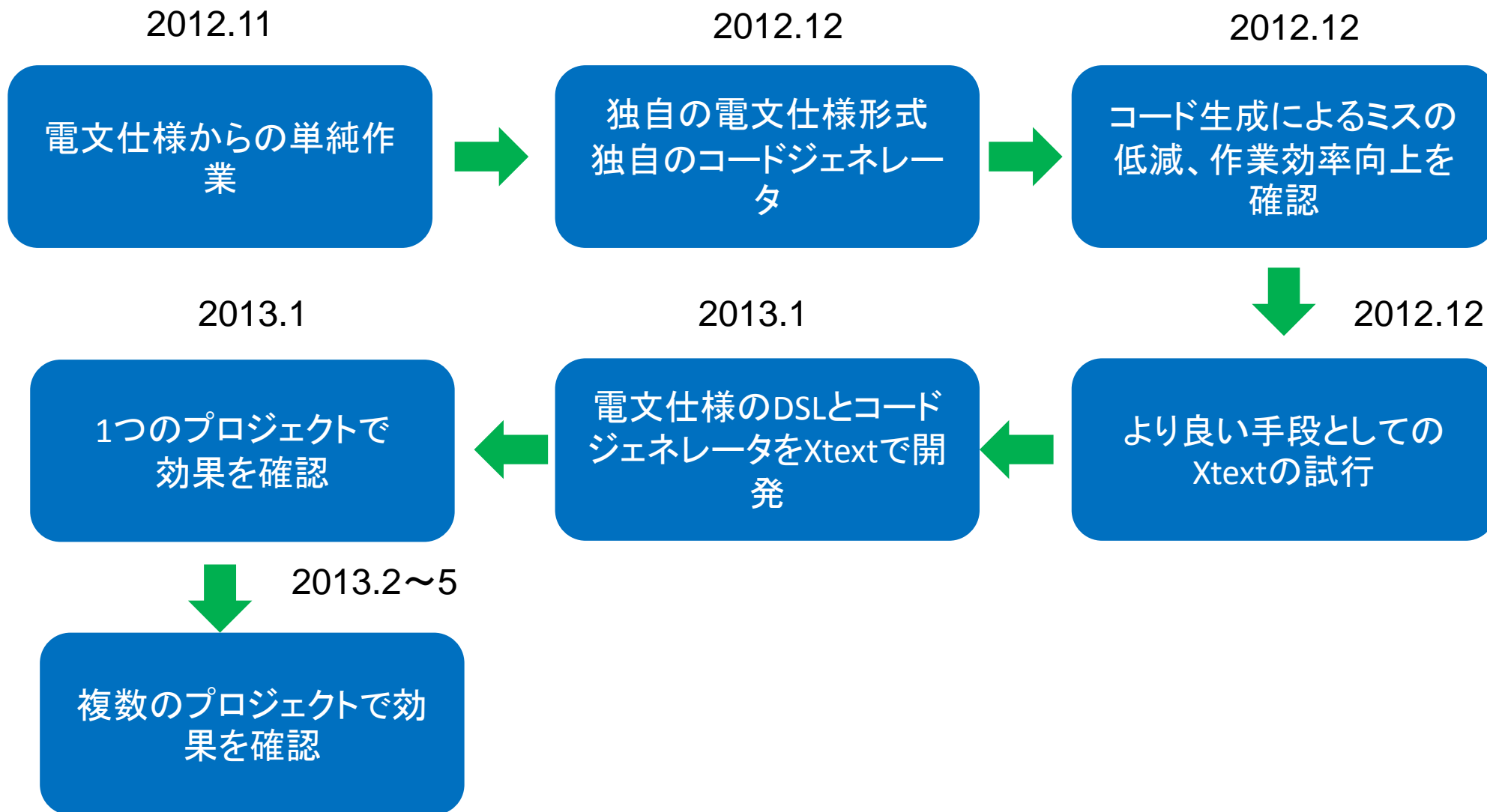
事例 DSLを用いたコード生成

活動	実施内容
情報収集	テキスト型DSLのXtextについてはSNS、ブログ等で認知。
課題の発見	アジャイル開発で2週間の反復をしている状況で通信プロトコルの解析処理の実装が電文解析を見ながらの単純作業になっている。
課題と手段のマッチング	電文仕様を形式化してコード生成する手段とマッチング。XtextのようなDSLの適用対象であると認識。
手段の試行	Xtextをインストール。チュートリアルを試行。
手段の実践	まずは独自のコードジェネレータを作成してコード生成を実施。
結果の評価	電文仕様の量が多かったので、コード生成によるミスの低減、作業効率化を実現できた。 独自のコードジェネレータ、電文仕様の形式は対象のプロトコル仕様専用。

事例の変遷



事例の変遷



事例の変遷後の利点

- 対象のプロジェクトメンバー全員がDSLの定義、コード生成の基本的なスキルを習得。
- 仕様から単純に設計、実装が可能なのをDSL化できないかと考えるようになった。
- 想定できる不具合の原因(DSLの定義間違いなど)を特定しやすくなったので、レビューやテストの観点が明確になった。
- 複数の概念や責務を一つのクラスに押し込めるような設計が減少した。(DSL→コード生成の観点からはむしろ面倒になったから)

事例2 実験的にやめてみる

- チームがユニットテストに時間が掛かりすぎると
いう課題を持っていた。
- ユニットテストでのカバレッジの目標、「単体」への
こだわりなど。



一週間、ユニットテストを実験的にやめてみた。

やめて何が起こるか観察する

- やめたことで、品質がどうなったか？
- やめたことで、どれだけ負債が溜まったか？
- 負債を返すために最低限何をすればいいか？
- ユニットテストを「やめ続けたら」何が起こりそうか？
- どこまでやると自分達にとって適切なのか？

2010年～2014年でチームで取り組んだ新手法

(1) プロセス

スクラム、XDDP

(2) 手法

アジャイルの技術的なプラクティス (TDD、ペアプロ、CI)

原因結果グラフ

テスト設計の段階的詳細化

アジャイルインスペクション

外部DSL

テスト自動化

(3) ツール

Jenkins、Atlassian製品 (JIRA[®], Confluence[®], Stash[®])

R

Visual Studio[®] Team Foundation Server

Sphinx

Xtext

Friendly、SpecFlow

Chef、Vagrant、Docker

チームへのアプローチ

- 新しいことに対するチャレンジにネガティブなチームで実験的アプローチを実践することは困難。
- 実験的アプローチの実践を意識したチームビルディングが必要。

コミットメントフリーなチーム

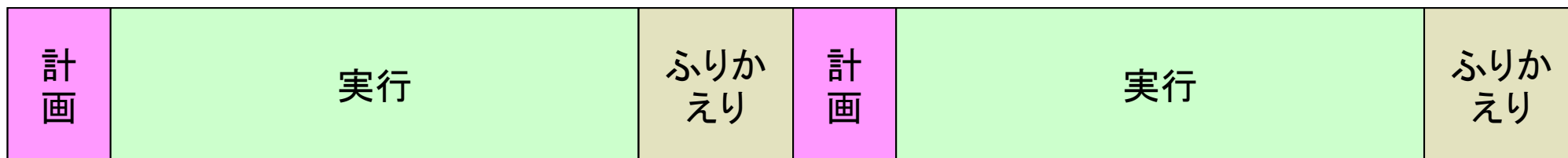
- チームにコミットメントを求めると、新しいことにチャレンジすることにネガティブになる。
- 「約束」のためには、確実な手段を必要とするから。
- コミットメントフリーにすることにより、チームにチャレンジを促す。
- 自分達が考えられる最大限改善を継続することにコミットしてもらおう。

チームへの提案

- チームに対して新しいチャレンジを提案する。
- 強制しない、ただし、議論をする。
- 最後はチームに決断を任せる。
- 同じような機会にまた繰り返し提案する。

実践への第一歩

- 作業に反復とふりかえりを取り入れる。



改善を提案する機会として「ふりかえり」を行う。

反復的に作業することにより、チャレンジの実施と評価、改善の機会を作る事ができる。