# Patterns of Developing Training and Involving People with Agile
## アジャイルな研修開発と人を巻き込むパターン

Ayana Chandler / Rakuten, Inc. / Agile Tech EXPO Organizer / Agile Japan Executive Committee
Kotaro Ogino / Rakuten, Inc.
Yasuo Hosotani / Agile Tour Osaka Committee

---

When HR is in charge of developing technical training, they do not have the technical expertise related to the content. We propose pattern languages where HR involves internal engineers and uses agile methodology to produce and continuously improve effective training material that meets company needs.

Categories and Subject Descriptors: A.0. [General] (The ACM Computing Classification System)
General Terms: Developing Training, Involving People

Additional Keywords and Phrases: Agile, Scrum, Fearless Change, HR, Bulldozer

---

## はじめに

人事が技術研修開発をする場合、研修対象の技術に関する専門知識がないために外部委託に研修を行ってもらう場合が多い。しかし、外部委託を利用すると、研修内容が委託先に依存してしまい、自社の状況に合った内容にしづらいという問題がある。また、技術の変化は速いため、研修もそれに順応して変化できるよう考慮しなければならない。

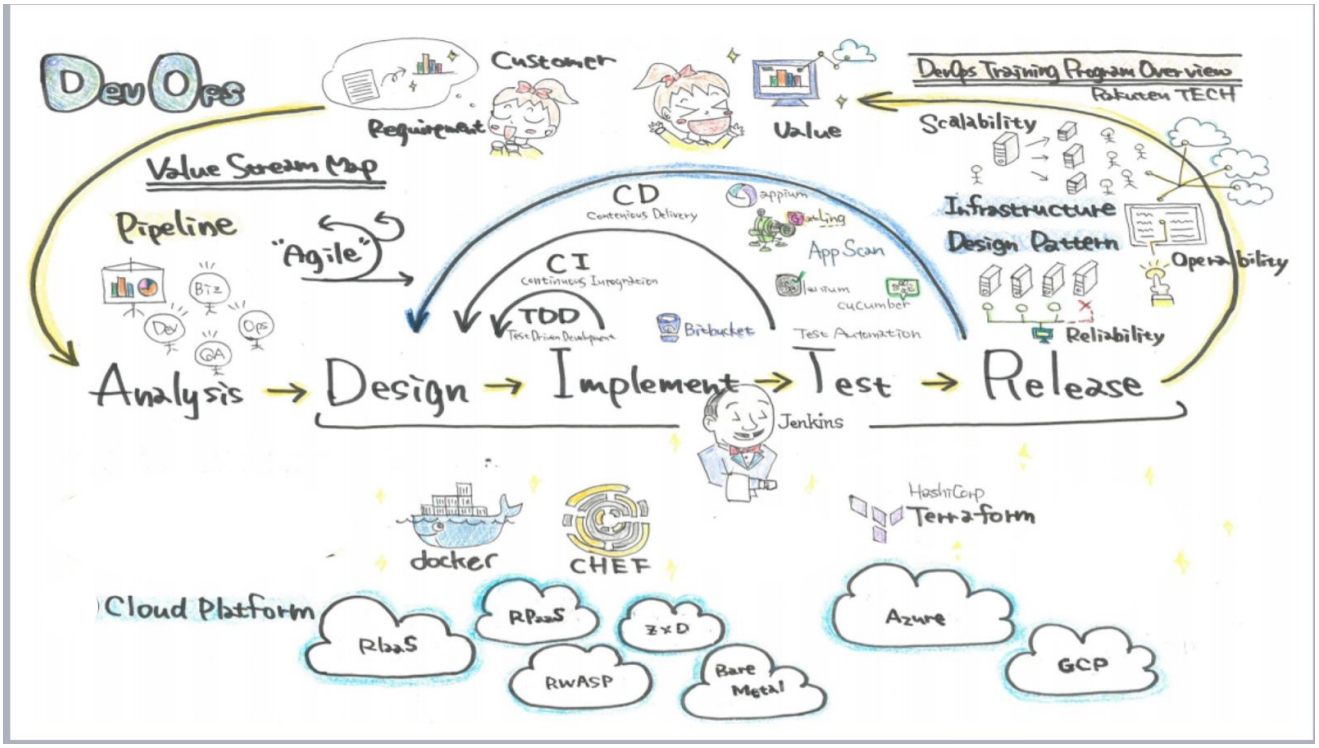
本論文では、人事が社内のエンジニアを巻き込み、アジャイルの手法で自社のニーズに合わせた効果的な研修を開発し、継続的に改善するためのパターンランゲージを提案する。

---

人事である筆者はこのパターンランゲージを用いて、社内で最も需要が高かったDevOps研修プログラムで、40人のエンジニアを研修開発に巻き込み、半年間で22個の研修開発に成功した。その半年間に32組織約300名の社内エンジニアがDevOps研修に参加し、エンジニアたちのフィードバックを元に研修の改善を繰り返した結果、NPS(ネットプロモータースコア)の中央値が6から31まであがった。

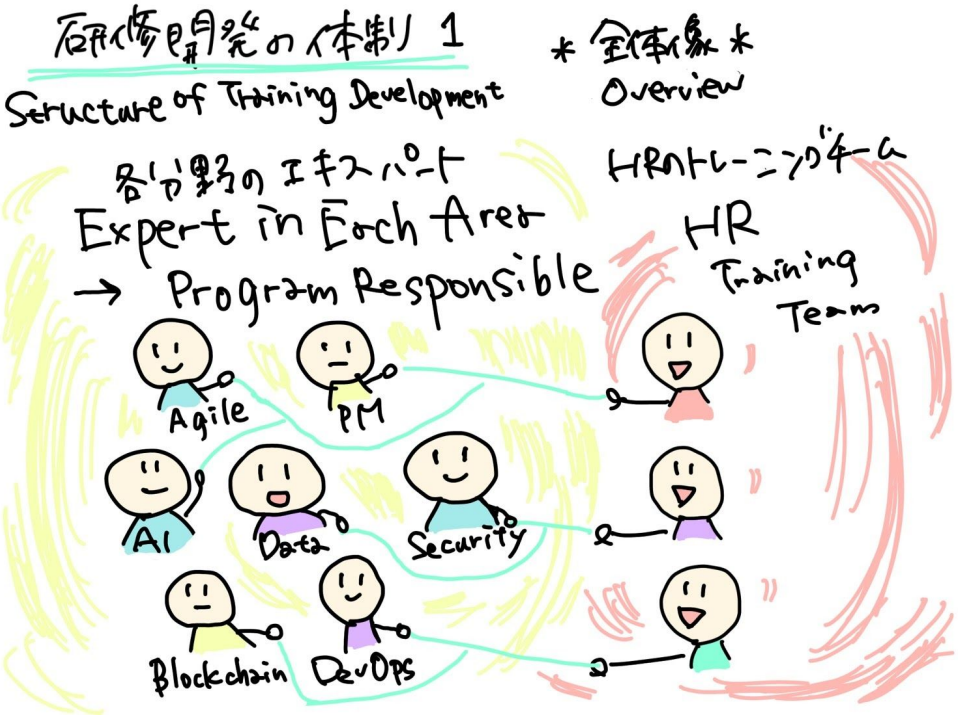
次の図は、ともに研修を作り上げたエンジニアの考えを元に、人事である筆者がDevOps研修プログラムのアイディアを図式化したものである。「プロダクトをアジャイルに開発し、顧客に最大限に価値を届ける」を根底に、DevOps研修プログラムは開発された。図には、プロダクト開発の流れと、開発に必要な技術やフレームワークが描かれ、それぞれが各研修のキーワードになっている。研修のカテゴリーは「パイプライン」、「インフラデザインパターン」、「クラウドプラットフォーム」の３つに分けられた。各カテゴリー内に、項目やレベルに分けて5〜10の研修が開発された。

本論文では、各研修内容の詳細ではなく、アジャイルを取り入れた研修開発の手法と仲間の巻き込み方について紹介していく。

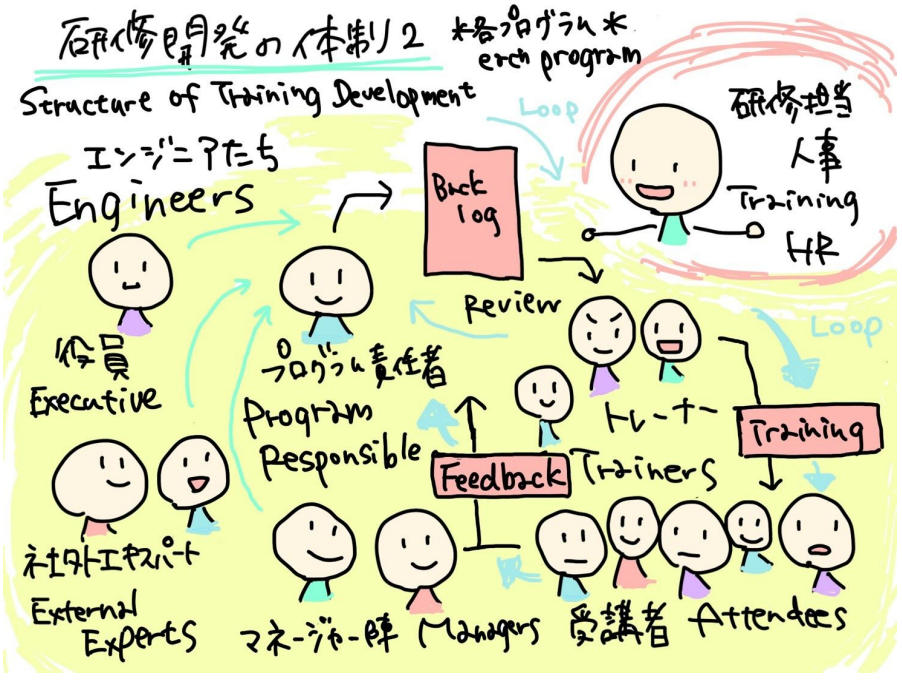


## 2. 研修開発の体制

ここでは、筆者の研修開発の体制を紹介する。




技術研修を開発していくにあたり、役員レベルで約10種類の必要な分野が定義された。社内で選定された各分野のエキスパートをプログラム責任者とし、HRのトレーニングチームのメンバーがプログラム責任者をそれぞれサポートする形で研修が企画運営された。



研修開発に関わる人々を、「研修開発チーム」と呼ぶ。アジャイル開発に用いられる「スクラムガイド」における「スクラムチーム」にも置き換えた言葉を添えながら、各役割を紹介していく。

◆ プログラム責任者(スクラムガイドにおけるプロダクトオーナー)

必要とされている研修を見極めそのトピックにおける研修プログラム全体をデザインする人を現場のエンジニアからアサインする。DevOps、AI、セキュリティーなど、分野ごとにそれぞれの専門家が担当し、各プログラムの中に複数の研修を開発する。
技術力が高く、社内の事情も世の中の技術の流行にも詳しい人物であることが望ましい。

◆ トレーナーチーム(スクラムガイドにおける開発チーム)

プログラム責任者が提示する研修を開発し、実施する。基本的には研修ごとにトレーナーは別のエンジニアになるが、同一人物が複数の研修を担当することもある。
研修内容における専門性が高く、トレーナーとしてのプレゼン能力やファシリテーション能力がある人物であることが望ましい。

◆ 研修担当人事(スクラムガイドにおけるスクラムマスター)

研修プログラムが滞りなく開発されるように、プログラム責任者とトレーナーチームを支援する。定例会議の実施やステークホルダーとの調整をする。組織に研修を広める。
多くの人と関わり、同時に調整をしていく必要があるため、コミュニケーション能力やマネジメント能力のある人物が望ましい。

3. パターン一覧

本論文にて提案するパターンの一覧を以下に示す。「Fearless Change」のパターンが研修開発に応用されているものが多いため、関連パターンを併記する。

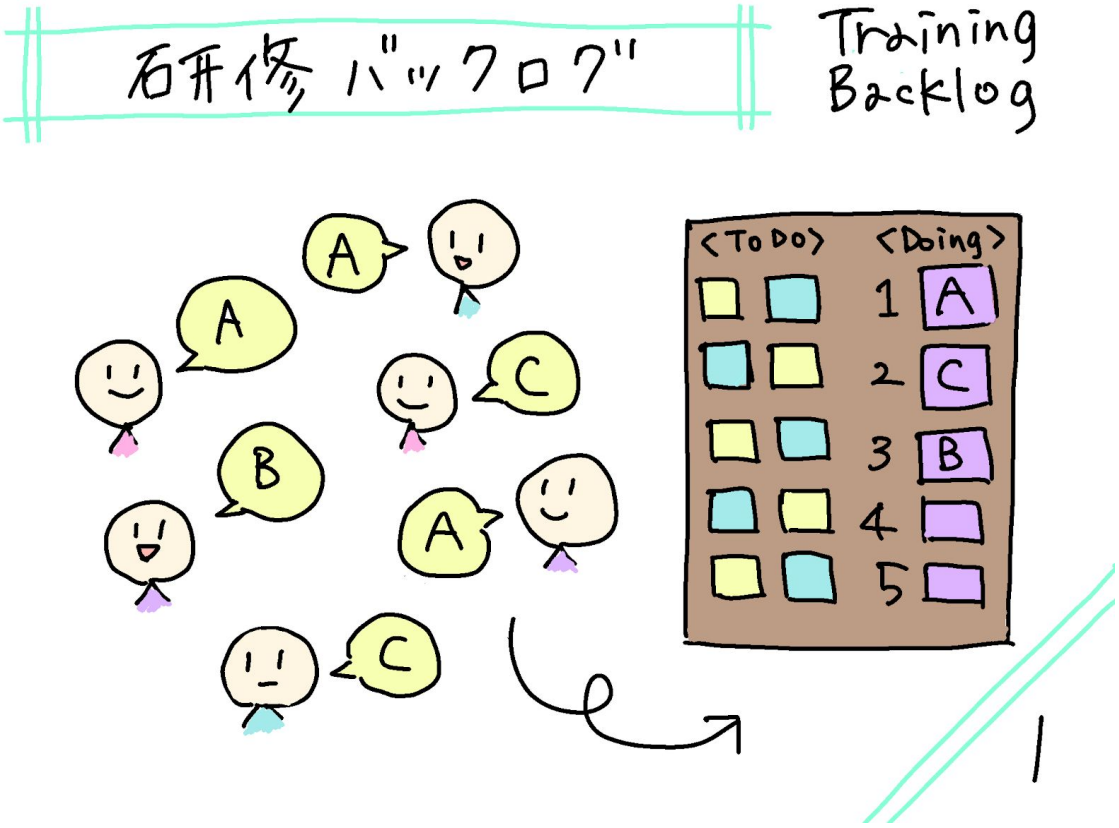
「アジャイルな研修開発と人を巻き込むパターン」は全体で大きく以下の２つに分類される。
◆パターン1〜7: アジャイルな研修開発パターン
◆パターン8〜18: ブルドーザーで人を巻き込むパターン

| パターン名 | パターンの要約 | Fearless Change<br>関連パターン |
|---|---|---|
| アジャイルな研修開発パターン | | |
| 研修バックログ<1> | 開発する研修のバックログを作ると、チームで優先順位の共通認識が持てる。 | ステップバイステップ(3)、予備調査(4)、やってみる(17) |
| 完成前トライアル<2> | 全部完成する前に小さくトライアルを実施すると、受講者のニーズに合っているか確認できる。 | 小さな成功(2)、ステップバイステップ(3)、予備調査(4)、やってみる(17)、お試し期間(47) |
| 定量的フィードバック<3> | アンケートに定量的な評価方法を用いると、全研修と比較できる。 | なし |
| みんなで振り返り<4> | 研修実施後は、毎回研修開発チーム全員で振り返りを実施し、良い点と改善点を見える化する。 | 小さな成功(2)、ふりかえりの時間(5) |
| 改善ダッシュボード<5> | 改善策をダッシュボードにまとめると、後から見返せるし、他のチームにも共有できる。 | 次のアクション(9)、体験談の共有(32) |
| 再構築する勇気<6> | 研修プログラム全体の改善が必要なら、思い切ってプログラム全体を再構築することもあっていい。 | なし |
| 柔軟カスタマイズ<7> | チーム単位で研修を依頼されたら、そのチームに合った研修にカスタマイズし、効果を上げる。 | テイラーメイド(26) |

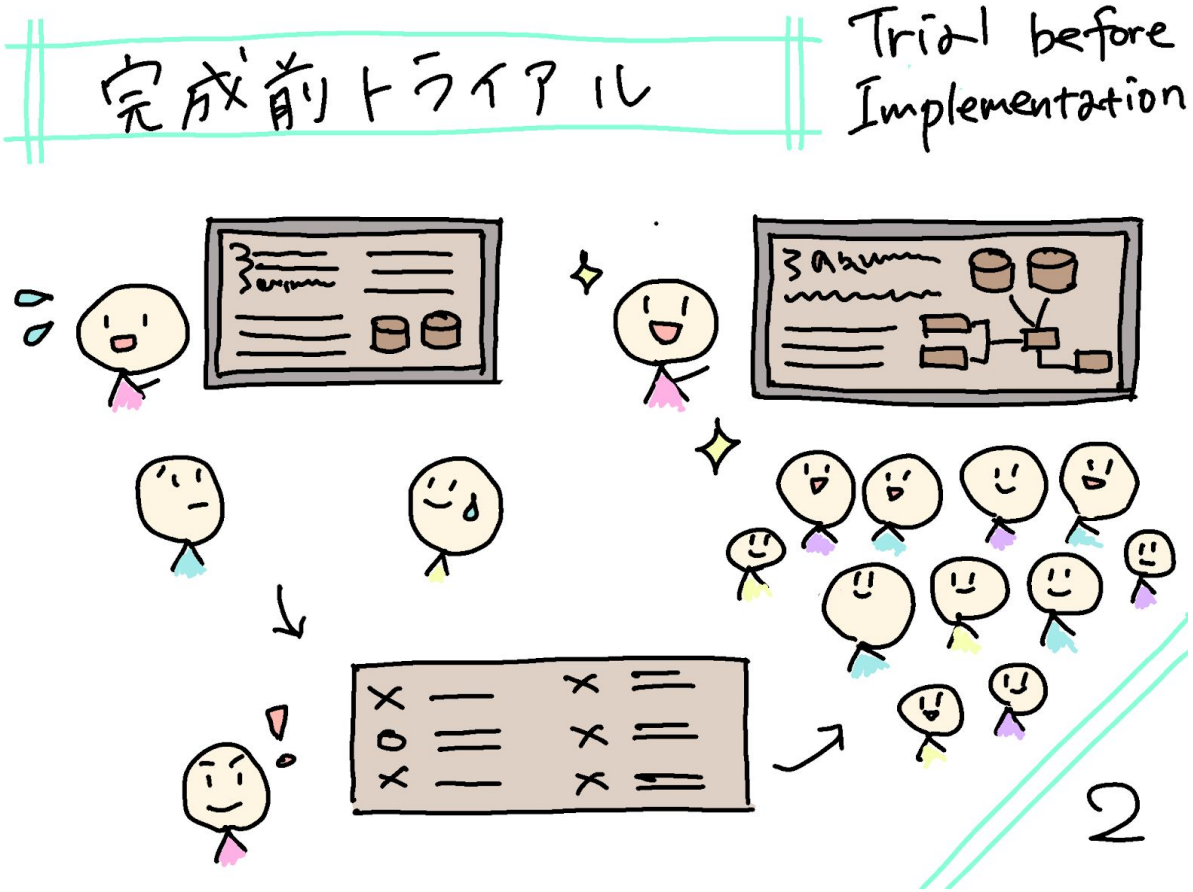
| ブルドーザーで人を巻き込むパターン | | |
|---|---|---|
| 人を巻き込むブルドーザー<8> | 人をたくさん巻き込むと、同時進行で複数の研修を開発できる。 | エバンジェリスト(1)、協力を求める(6)、みんなを巻き込む(33) |
| 構わず突進<9> | 積極的に話しかけ熱意を伝えると、人は協力してくれるし心を開いてくれる。 | 協力を求める(6)、みんなを巻き込む(33)、恐れは無用(46) |
| 無邪気な無知<10> | 自分が知らないことを相手に隠さないでいると、教えてもらえるし、信頼関係も築かれる。 | 協力を求める(6)、感謝を伝える(18)、恐れは無用(46) |
| 愛される自己主張<11> | 知識がなくても議論に関わるには、自分なりの視点で意見を伝えることが大切だ。 | 体験談の共有(32) |
| 相方絶対論<12> | お互いの苦手な部分を補い合い、心の支えになる相方がいると強い。 | 協力を求める(6)、コネクター(8)、達人を味方に(14)、イノベーター(16)、達人のレビュー(31)、メンター(37)、相談できる同志(39) |
| イノベーター発掘<13> | 新しい技術を積極的に導入している人には、最新の技術やベストプラクティスを研修で広めてもらう。 | コネクター(8)、イノベーター(16)、勉強会(25)、橋渡し役(43) |
| ぼっちにさせない<14> | トレーナーにも相方やチームを作り、協力し合いながら研修開発をしてもらう。 | 協力を求める(6)、定期的な連絡(24)、みんなを巻き込む(33)、相談できる同志(39) |
| 組織の壁打破<15> | 部署やレイヤーを越えてトレーナーチームを作ると、研修以外のことも情報交換し合うようになる。 | みんなを巻き込む(33)、橋渡し役(43) |
| 専門家のアドバイス<16> | 社外の専門家からなら、トレーナーもアドバイスをもらいやすい。 | 外部のお墨付き(12)、達人を味方に(14)、達人のレビュー(31)、みんなを巻き込む(33) |
| トレーナーベネフィット<17> | トレーナーにしか体験できない場を提供し、やりがいを感じてもらう。 | 何か食べながら(9)、グループのアイデンティティ(13)、感謝を伝える(18)、個人的な接触(20) |

| | | |
|---|---|---|
| プロモーターを味方に<18> | SNSでの宣伝やイベントの開催を通し、新しいもの好きな人たちを味方につけ、支持者を増やす。 | アーリーアダプター(11)、種をまく(22)、勉強会(25)、著名人を招く(27)、経営層の支持者(28)、アーリーマジョリティー(30)、みんなを巻き込む(33)、将軍の耳元でささやく(48) |

4. パターンの説明
以下にそれぞれのパターンについて説明する。
◆パターン1～7: アジャイルな研修開発パターン




研修バックログ<1>
1. 状況
　研修開発チームのもとには、エンジニアや役員から要望が届き、プログラム責任者はその要望や社内外の動向を考慮して、開発すべき研修を決める。
2. 問題
　複数の研修を作成するため、数が増えると開発状況の管理がしづらい。
3. フォース
　「教えた方がいい」という観点で研修の数は増加するが、技術の変化は速いため、全研修が完成する頃には先にできたものが陳腐化することが予測される。
4. 解決方法
　研修担当人事は、プログラム責任者の考えにもとづき開発すべき全研修を書き出し、バックログ(案件リストのようなもの)を作成する。アジャイルのカンバンのように、優先順位や開発進捗に従ってバックログを動かし、誰でも見れるようにしておく。優先度の高い研修から開発し、完成した研修から順次実施する。
5. 結果状況
　研修開発チーム全員が同じバックログを見るため、優先順位に共通の認識が持てる。完成したものから実施するため、陳腐化することがなく、プログラム開始の初期段階から受講者のフィードバックも集められる。




完成前トライアル<2>

1. 状況

　研修開発チームが研修を開発している間は、受講者の反応が見れない。

2. 問題

　トレーナーチームが時間をかけてニーズに合わない研修を開発していたことが、研修が完成して実施され、受講者の反応を見てわかることがある。

3. フォース

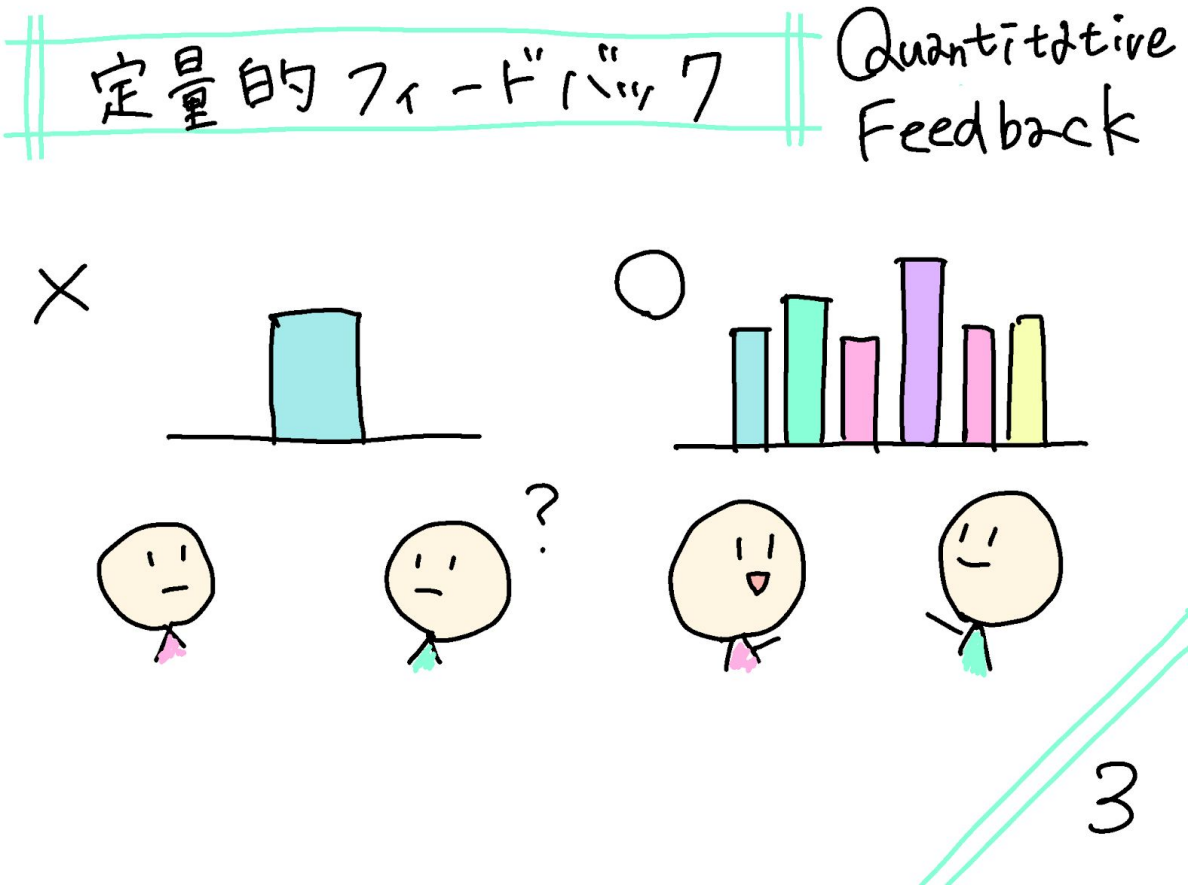
　受講者もニーズに合った研修が欲しいので、完成前に要望を伝えたい。

4. 解決方法

　トレーナーチームは、研修が全て完成する前に、開発途中の小さい範囲でトライアル研修を実施する。トレーナーの所属部署などその分野の専門家が多いチームや、将来的に受講者となる可能性の高いチームを対象に声をかけ、トライアル研修に参加してもらい、フィードバックをもらう。

5. 結果状況

　トライアル研修受講者のフィードバックから良い点と改善点が見え、その研修をより良い状態で完成できる。




定量的フィードバック<3>

1. 状況

　完成した研修は繰り返し実施され、毎回受講者のフィードバックを得るためにアンケートをとる。

2. 問題

　研修ごとに異なるアンケートを作成してしまうと、他の研修との比較が難しい。

3. フォース

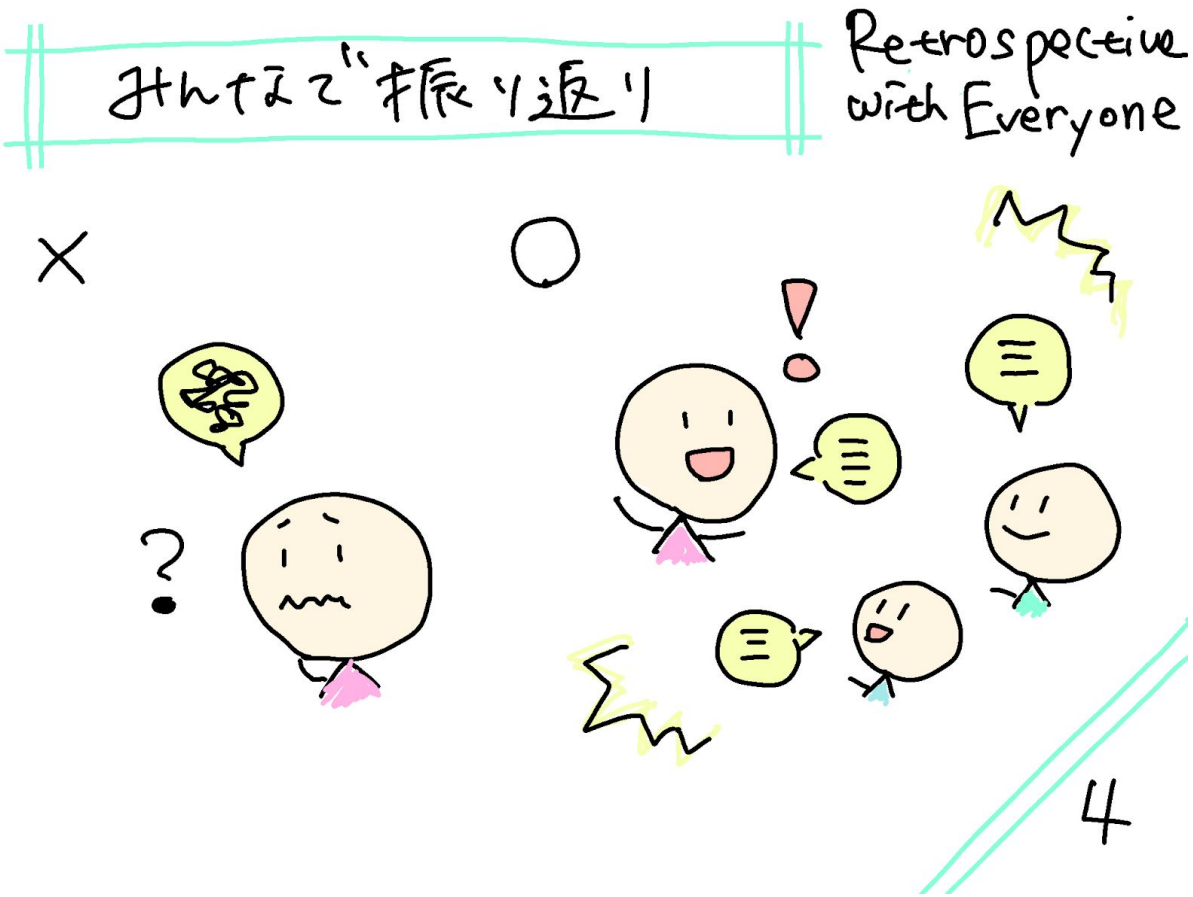
　受講者アンケートはコメントなど定性的な評価だけを参考にしてしまうことが多い。

4. 解決方法

　NPS(Net Promoter Score) など、定量的な評価方法を用いてアンケートを実施する。また、全研修で毎回同じ内容のアンケートを実施する。

5. 結果状況

　全て同じ内容のアンケートで、数値で結果を出すため、同じ研修の前回との比較だけでなく、他の研修との比較もできる。全研修で比較ができるため、どのような研修が受講者満足度が高いのかを分析しやすくなり、満足度の高い研修の良い点を他の研修に取り組むことができる。数値が出るため、全研修の目標値を決めることもできる。




みんなで振り返り<4>

1. 状況

　完成した研修は繰り返し実施され、定量的フィードバック<3>を参考に、必要があれば毎回改善をする。

2. 問題

　定量的フィードバック<3>を参考にするものの、トレーナーチームだけの視点では、改善方法が限られてしまう。

3. フォース

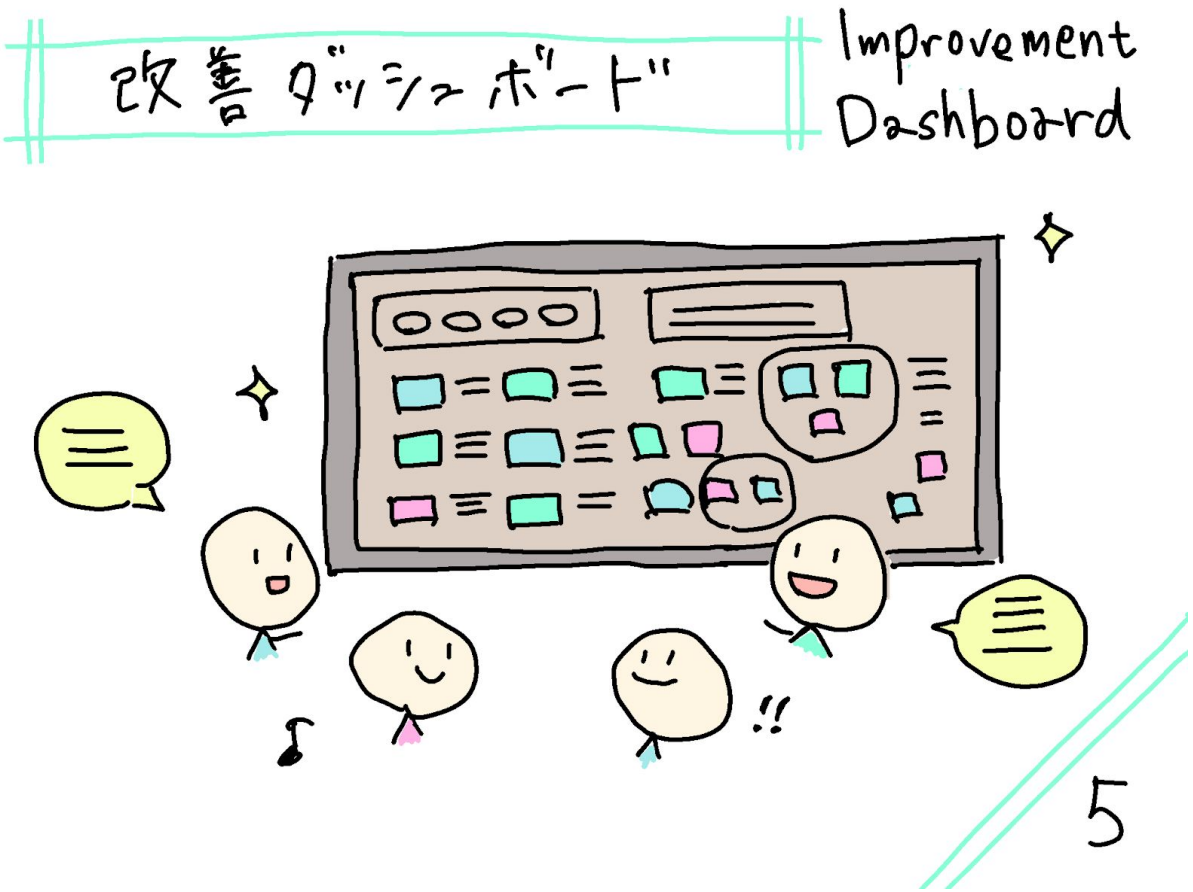
　違う視点での意見や複数人の合意があれば、改善方法が明白になりやすい。

4. 解決方法

　研修実施後は、定量的フィードバック<3>を元に、研修担当人事もプログラム責任者も含む研修開発チームがみんなで振り返りミーティングをし、継続すべき点と改善すべき点を議論し、参加者全員の合意の元に改善点のバックログを作る。次回までに改善できるように、〆切や担当者を決める。

5. 結果状況

　複数の立場が集まり、あらゆる角度から改善のアイディアが出せる。例えば、研修担当人事は他の研修の事例も知っているため、トレーナーチームにない情報を提供できる。また、全員で改善点のバックログを作成するため、次回までに改善することに対して責任感が生まれる。この繰り返しにより、研修の質が毎回あがる。




改善ダッシュボード<5>

1. 状況

　毎回研修後のみんなで振り返り<4>で継続すべき点と改善点があげられる。

2. 問題

　研修開発チーム全員は改善点について同じ認識を持っているべきだが、量が多いと認識にずれが生じることがある。

3. フォース

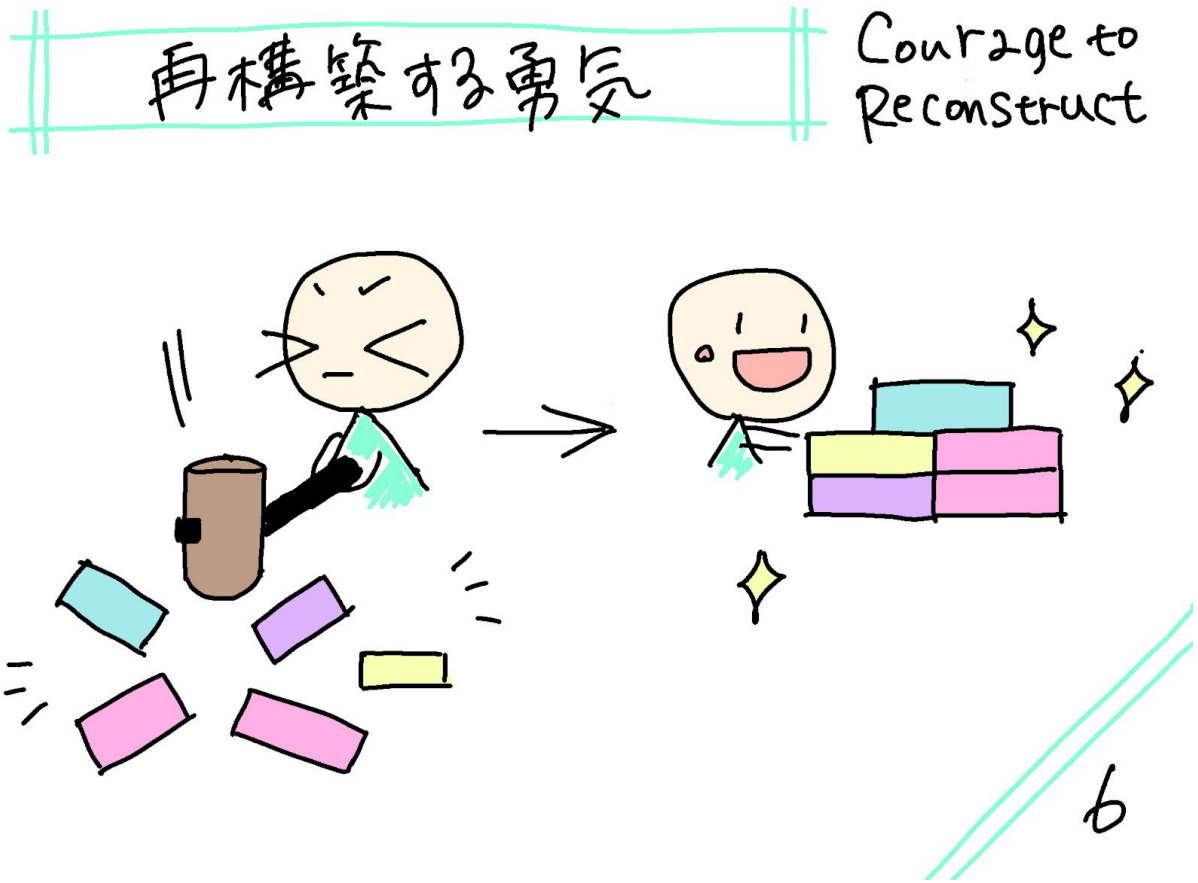
　特にプログラム責任者と研修担当人事は他にも沢山の研修を見ているため、情報が錯綜する。

4. 解決方法

　研修担当人事は、全研修の毎回のみんなで振り返り<4>の内容を改善ダッシュボードにまとめ、議論した継続すべき点や改善点を見える化する。

5. 結果状況

　研修開発チーム全員が同じ認識を持ちやすくなる。また、他の研修のトレーナーにも共有しやすくなり、良い改善方法が広まる。




再構築する勇気<6>

1. 状況

　完成した研修をいくつか実施したが、複数の研修の定量的フィードバック<3>とみんなで振り返り<4>の結果、研修プログラム全体の構造が間違っていたのではないかという疑いができる場合がある。

2. 問題

　小さな修正を繰り返して続行しても、研修の効果の改善にならない。

3. フォース

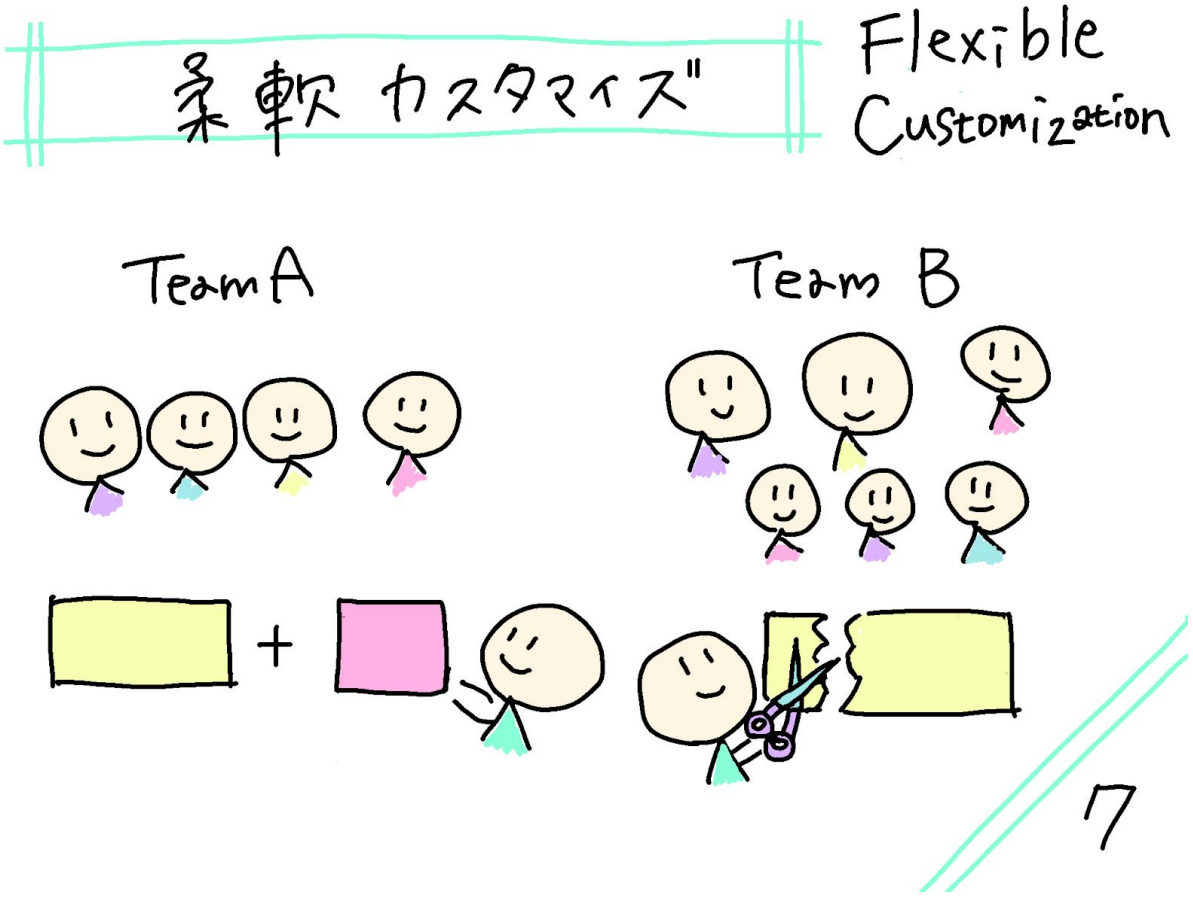
　すでに複数の研修が完成しているため、抜本的な変更やプログラム全体の再構築はしづらい。

4. 解決方法

　プログラム責任者は、定量的フィードバック<3>とみんなで振り返り<4>が示している受講者や関係者の言葉とデータをより重視し、研修プログラム全体の再構築を行う。

5. 結果状況

　失敗してしまった事実がマイナスに見えるのははじめだけで、研修プログラム全体が見直されたことでより効果のある研修が実施でき、受講者満足度が全体的にあがる。定量的フィードバック<3>とみんなで振り返り<4>でも良い結果が出る。




柔軟カスタマイズ<7>

1. 状況

　１つのチームから、チーム全員で研修を受講したいとの依頼がある。

2. 問題

　既存の研修をそのまま提供するだけでも知識やスキルは学べるが、せっかくチーム全員で受講するのに、それが活用できていない。

3. フォース

　同じチームだと課題を共有しているため、より深い議論ができる。チーム全員が一緒に仕事から離れて話し合うことができる機会なので、工夫をすれば、そのチームが抱えている課題を解決する時間にできるかもしれない。

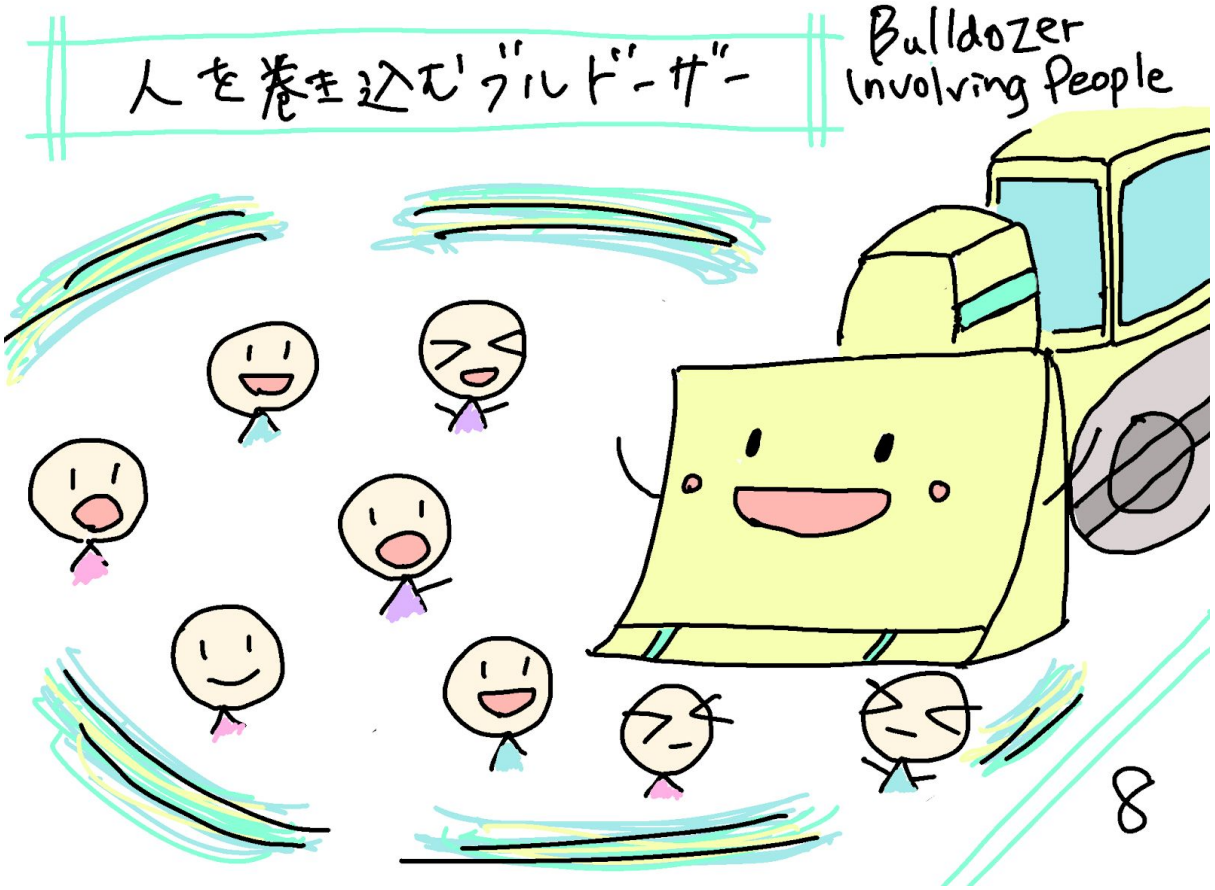
4. 解決方法

　研修担当人事が事前にチームの状況や研修を通して改善したい点をヒアリングし、チームの状況に合わせて研修をカスタマイズすることを研修開発チームに提案する。別の研修を組み合わせたり、内容を変更したりすることにより、チームが抱えている問題を解決できないか探る。

5. 結果状況

　チームの状況に合った研修内容にすることで、研修を通してチームが抱えている問題を見つけ出したり解決することができ、チームにとって研修がより意味のある時間となる。

◆パターン8～18: アジャイルに人を巻き込むパターン




人を巻き込むブルドーザー<8>

1. 状況

　研修担当人事は技術の専門知識がないので、自分だけでは技術研修を作成することができない。

2. 問題

　社外の専門家に講師を依頼することもできるが、その場合は自社の状況が的確に伝わらず、内容を合わせてもらいづらい。社内のエンジニアに講師を依頼しようとすると、講師ができるエンジニアは任されている業務も多く、研修開発に時間を割くことができない。

3. フォース

　トレーナーが一人だと負担が集中するが、複数人であれば負担が分散する。

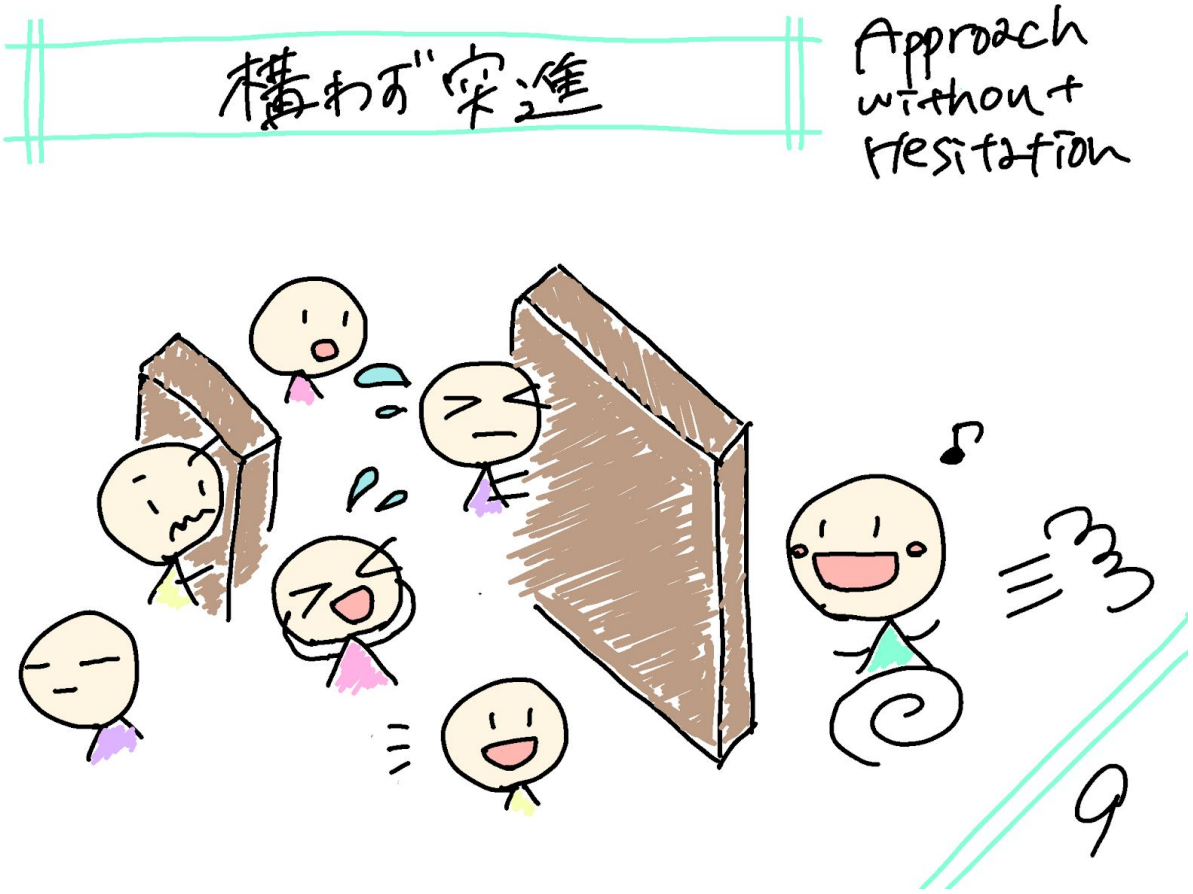
4. 解決方法

　後出のパターンを用いて社内外両方のエンジニアを複数名巻き込み、役割を分担したり協力し合いながら、同時進行で複数の研修開発を進行する。

5. 結果状況

　同時進行で複数のチームが動いたことで、短期間で複数の技術研修が完成する。また、社外の専門家と社内の事例を知っているエンジニアが協力し合うことで、社内外の事例が組み込まれた研修ができる。

　人を巻き込む活動によりトレーナーだけでなく、支持者や協力者も増やる。




構わず突進<9>

1. 状況

　研修担当人事が技術研修を開発し受講者を増やすためには、エンジニアの中に協力者や支持者を増やしたい。

2. 問題

　人事とエンジニアは、違う職種のため通常であれば関わることが少なく、お互いを理解して関係性を構築するのが同じ職種同士より難しい場合が多い。

3. フォース

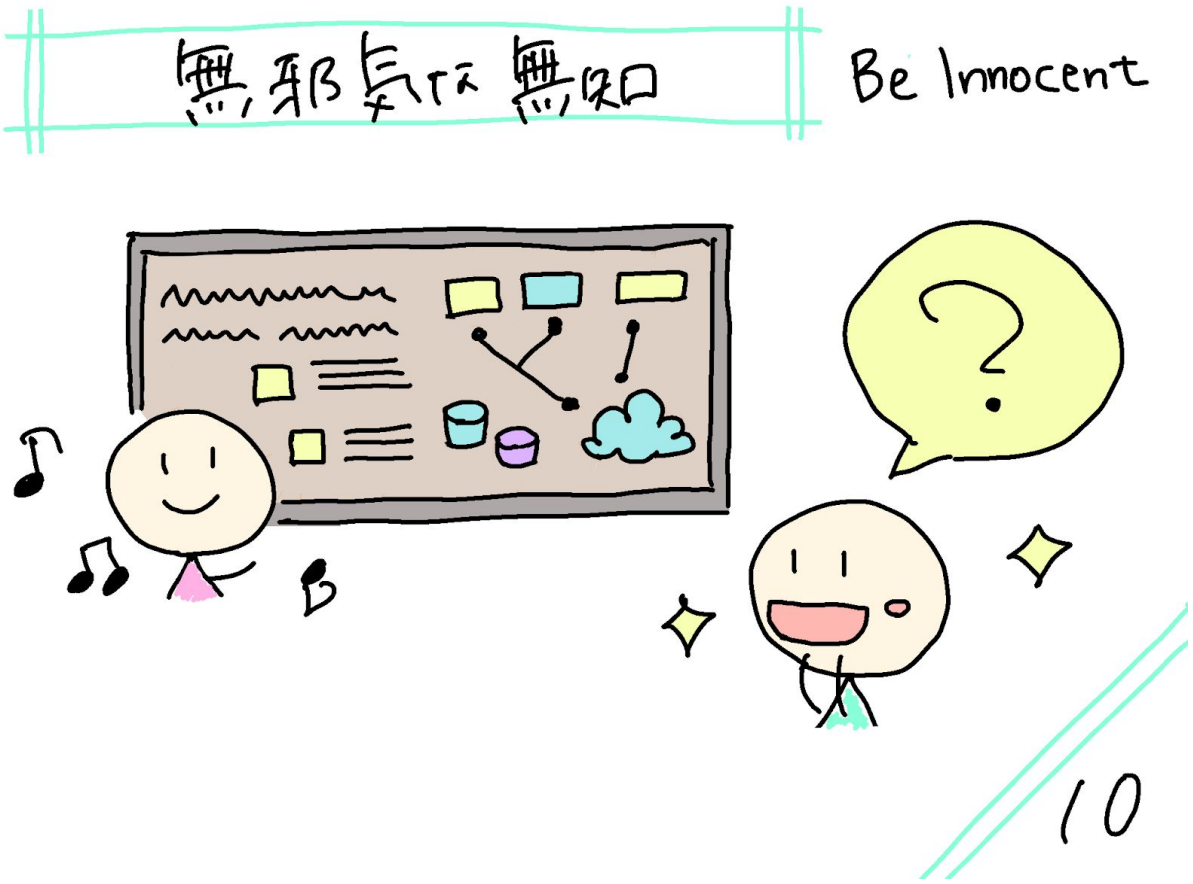
　エンジニアは通常業務が忙しく、研修開発に積極的に携わる余裕が時間的にも精神的にもあまりない場合が多い。

4. 解決方法

　研修担当人事は、エンジニアにそっけなくされても、積極的に話しかけ続け、実現したい具体的内容と熱意を伝える。冷たくされたとしても、伝わるまで何度も繰り返しアプローチする。

5. 結果状況

　積極的に話しかけ続けると、耳を傾けてくれるようになり、熱意が伝わり、協力してくれるようになる。心を開いてくれて、いろんな話をしてくれるようになる。




無邪気な無知<10>

1. 状況

　研修担当人事は、技術の専門知識がなく、エンジニアの技術の話がわからない。

2. 問題

　エンジニアと仲良くなりたいが、技術の話ができないので、話が盛り上がらない。

3. フォース

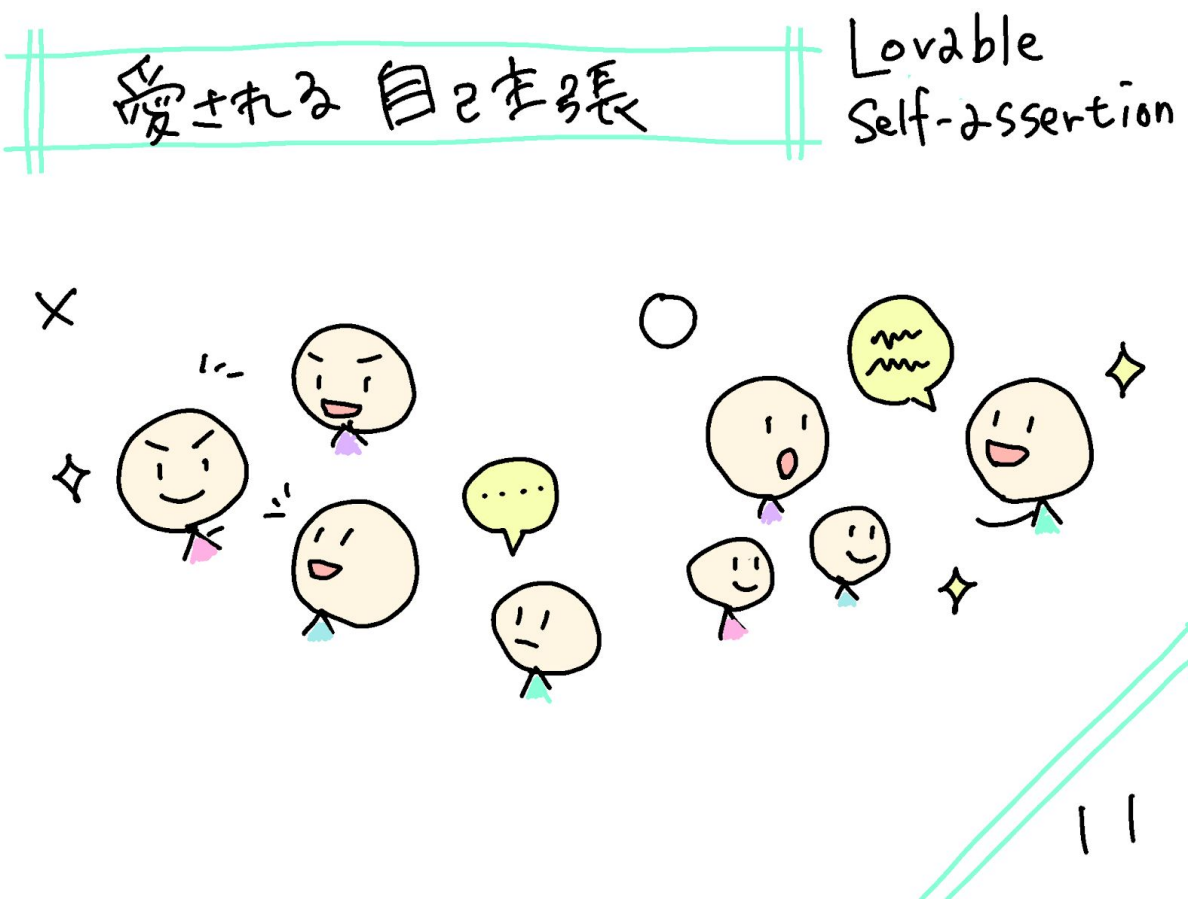
　人は知らないことを認めるのが恥ずかしい。 一方で、人は知っていることや好きなことを話すのは楽しい。

4. 解決方法

　研修担当人事は、恥ずかしくても、知らないことを隠さず素直に伝えるようにする。好奇心を示し、わからないことを質問する。

5. 結果状況

　エンジニアは技術の話をするのが好きなことが多く、質問をすると、楽しそうに教えてくれる場合が多い。その姿勢で会話を続けていくと、研修担当人事も最初はわからなかったことがわかるようになり、技術研修についてエンジニアと議論できるようになる。その姿勢が認められると、信頼してもらえるようにもなる。




愛される自己主張<11>

1. 状況

　研修担当人事は、技術の専門知識がないので、研修のコンテンツについてはエンジニアと対等に会話することができない。

2. 問題

　技術の知識がないからと研修内容を決める議論に参加しないと、エンジニアと距離ができてしまう。

3. フォース

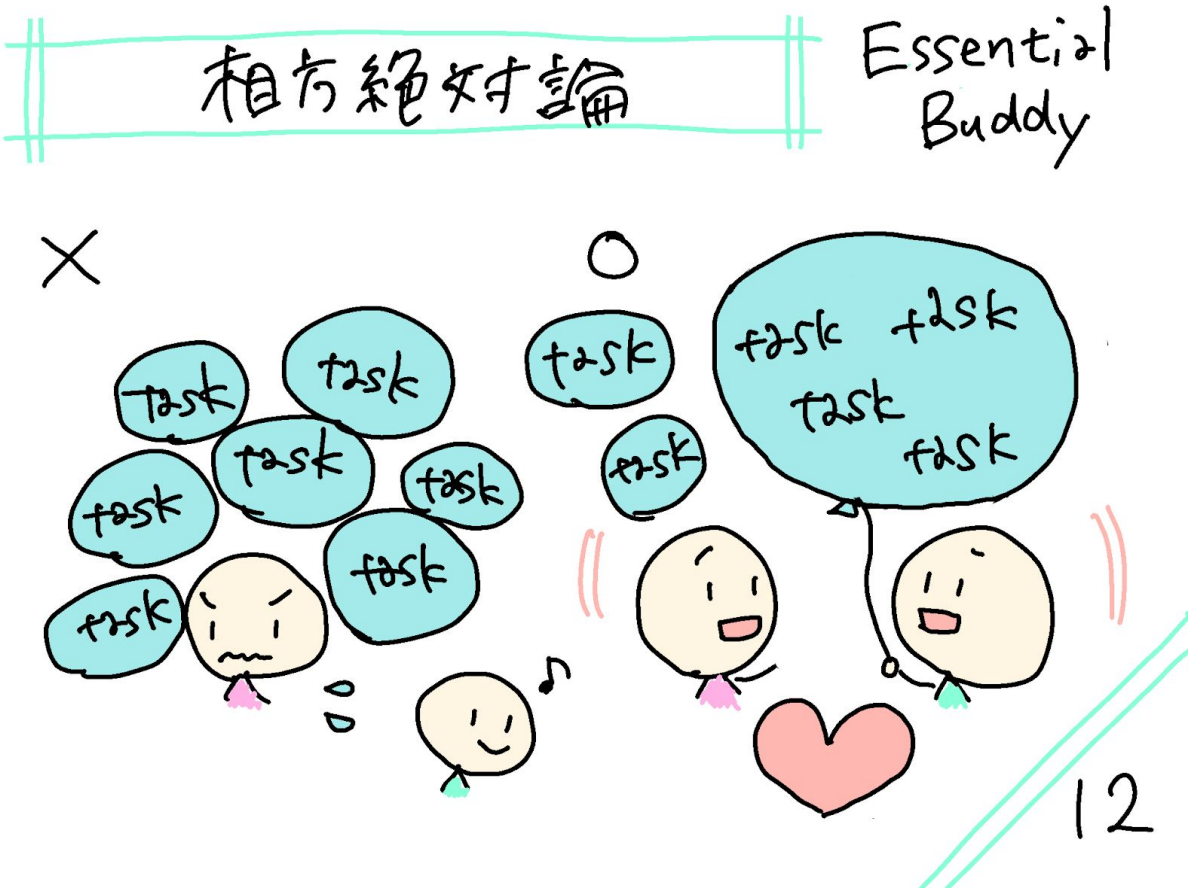
　自分より知識がある人に対して、自分の意見を主張するのは抵抗がある。

4. 解決方法

　研修担当人事は技術の知識がないことに気負いしなければ、エンジニアでは気づけない部分で、研修をより良くするアイディアを持っている。人前での話し方や受講者のケアの仕方など、技術以外の部分でも議論すべき点はたくさんあり、全研修を見続けている研修担当人事の方が多く体験している部分もある。研修担当人事はその事実に目を向け、積極的に議論に関わっていく姿勢を見せる。

5. 結果状況

　エンジニアだけの議論では出なかった新しい視点が追加され、研修がより良くなる。研修担当人事も開発に関わった感じが出て、研修開発チームに一体感が生まれる。




相方絶対論<12>

1. 状況

　研修担当人事は、技術の専門知識がないので、社内外の技術動向に詳しい専門知識のあるエンジニアをプログラム責任者におく。

2. 問題

　プログラム責任者だけに研修開発の企画や進捗管理、運営を全て任せきりにすることもできるが、プログラム責任者は通常任されている業務も多く、忙しい。

3. フォース

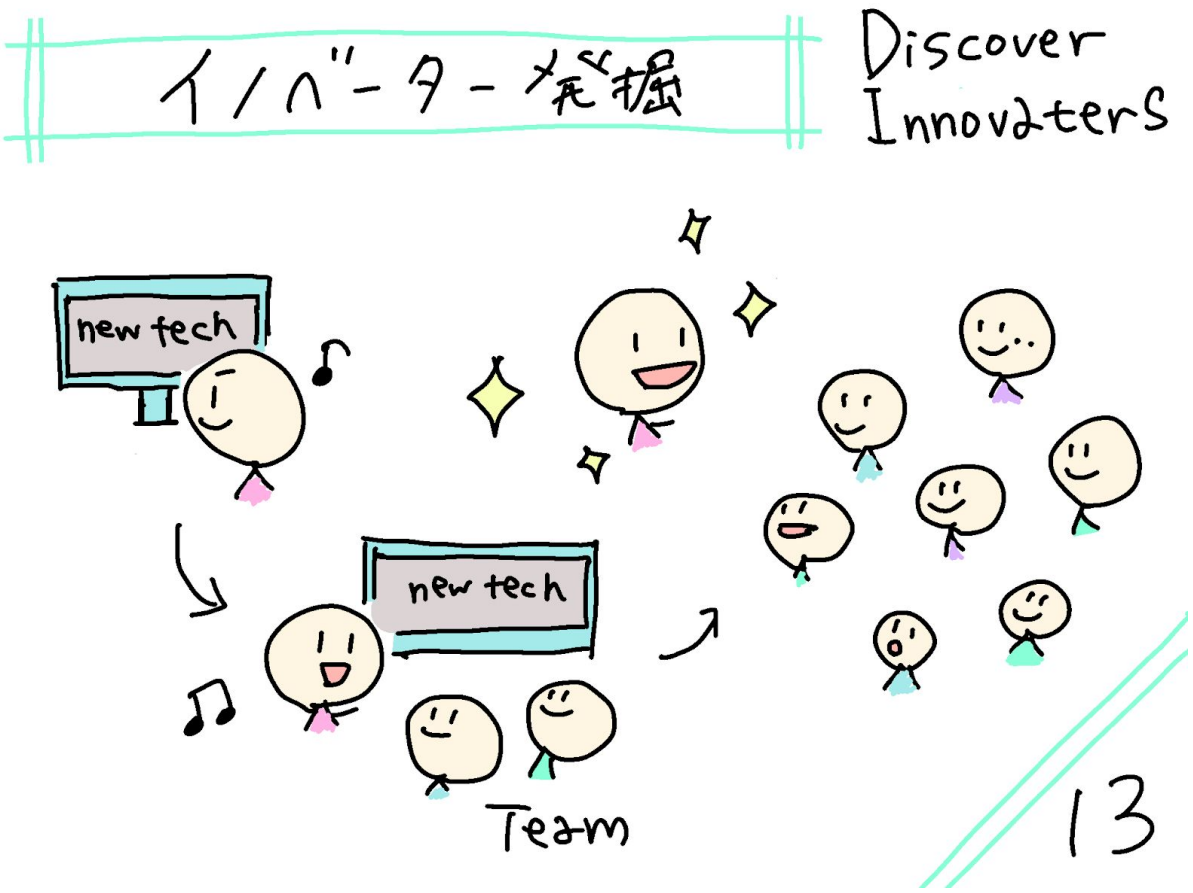
　プログラム責任者だけに任せきりにすると、手続きや調整などもあり、モチベーションがわかず、負担感も大きい。

4. 解決方法

　プログラム責任者が研修内容の企画に集中できるよう、研修担当人事はそれ以外の作業を率先して対応する。研修内容の企画以外には、資料作成、会議の設定と司会、役員や関係者との調整、トレーナーのスケジュール管理、研修場所の手配、受講者管理、アンケート管理などがある。

5. 結果状況

　研修担当人事が「できること全てする」姿勢でいると、プログラム責任者も手伝ってくれるようになる。プログラム責任者と研修担当人事が、相方同士のようにお互いに強力し合い、支え合う関係性ができると、研修開発がスムーズに進む。信頼関係が構築されると、人間関係などでトラブルが起こりそうな部分でも、お互いをフォローし合うようになる。




イノベーター発掘<13>

1. 状況

　研修担当人事とプログラム責任者は、研修を自ら開発しトレーニングを行えるトレーナーを探す。

2. 問題

　研修業務は、通常の開発業務に追加で行う業務となるため、エンジニアにとっては負担となる場合が多い。現場のマネージャーに指名してもらうと、即戦力ではないエンジニアが指名されることすらある。

3. フォース

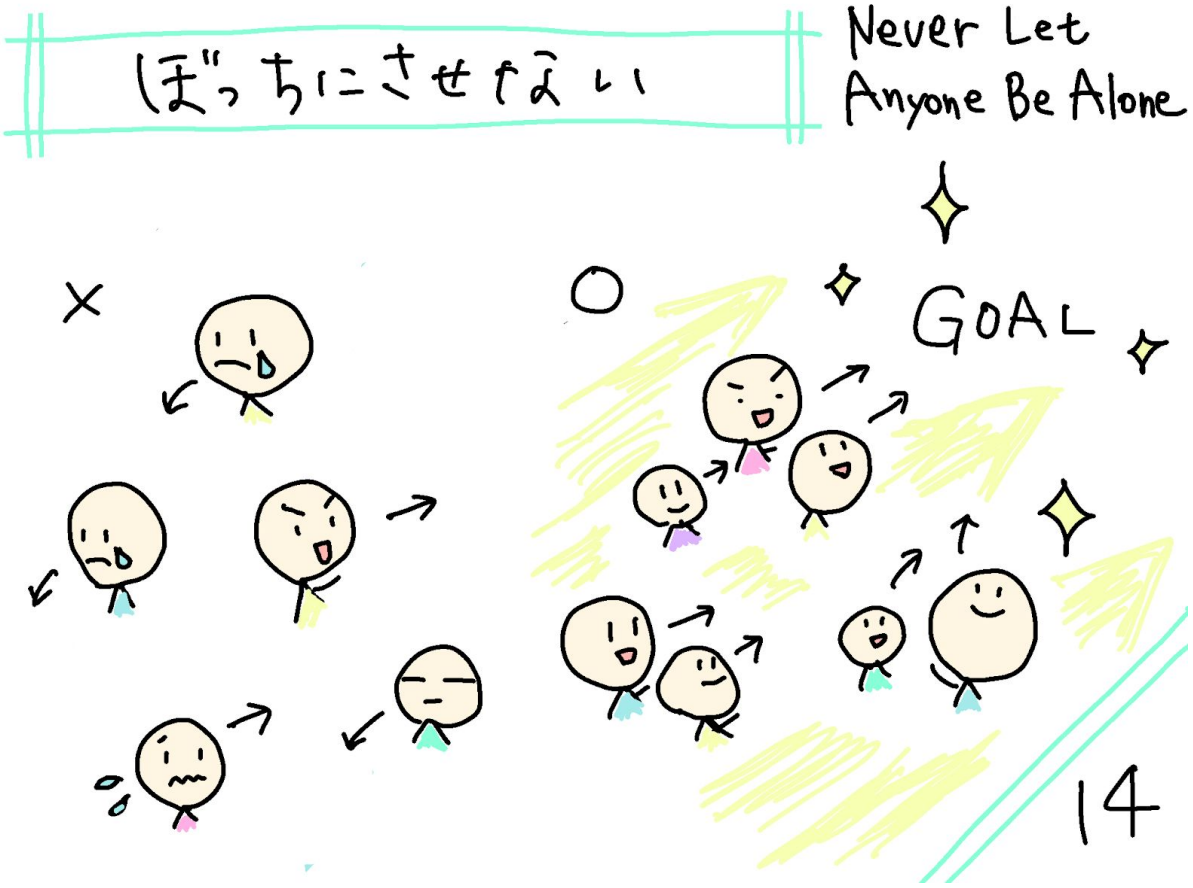
　新しい技術を積極的に導入している「イノベーター」と呼ばれるような人は、現場の即戦力でありエンジニアからも認められていて、トレーナーに向いている。

4. 解決方法

　プログラム責任者と繋がりのあるイノベーターに声をかけ、トレーナーを引き受けてもらえるよう本人と上司に依頼する。イノベーターは技術力が高いだけでなく、ベストプラクティスを共有することに興味がある人が多く、勉強会やSNSでの発信を積極的に行っているため、見つけやすい。

5. 結果状況

　技術力が高くかつ推進力のあるイノベーターがトレーナーになり、研修を通して最新の技術や社内のベストプラクティスが、広まるようになる。また、イノベーター同士繋がっていることが多いので、芋づる式に他のイノベーターも紹介してもらえ、人を巻き込むブルドーザー<8>でトレーナーチームが拡大する。




ぼっちにさせない<14>

1. 状況

　トレーナーは、研修開発チームの中でも研修の教材やハンズオン環境を作成する立場なので業務の負担が大きいが、通常業務をやりながらそれをこなさなければならない。

2. 問題

　１人でやると内容や負担が偏ってしまう可能性がある。また、忙しいときは通常業務が優先され、研修開発は後回しになってしまう。

3. フォース

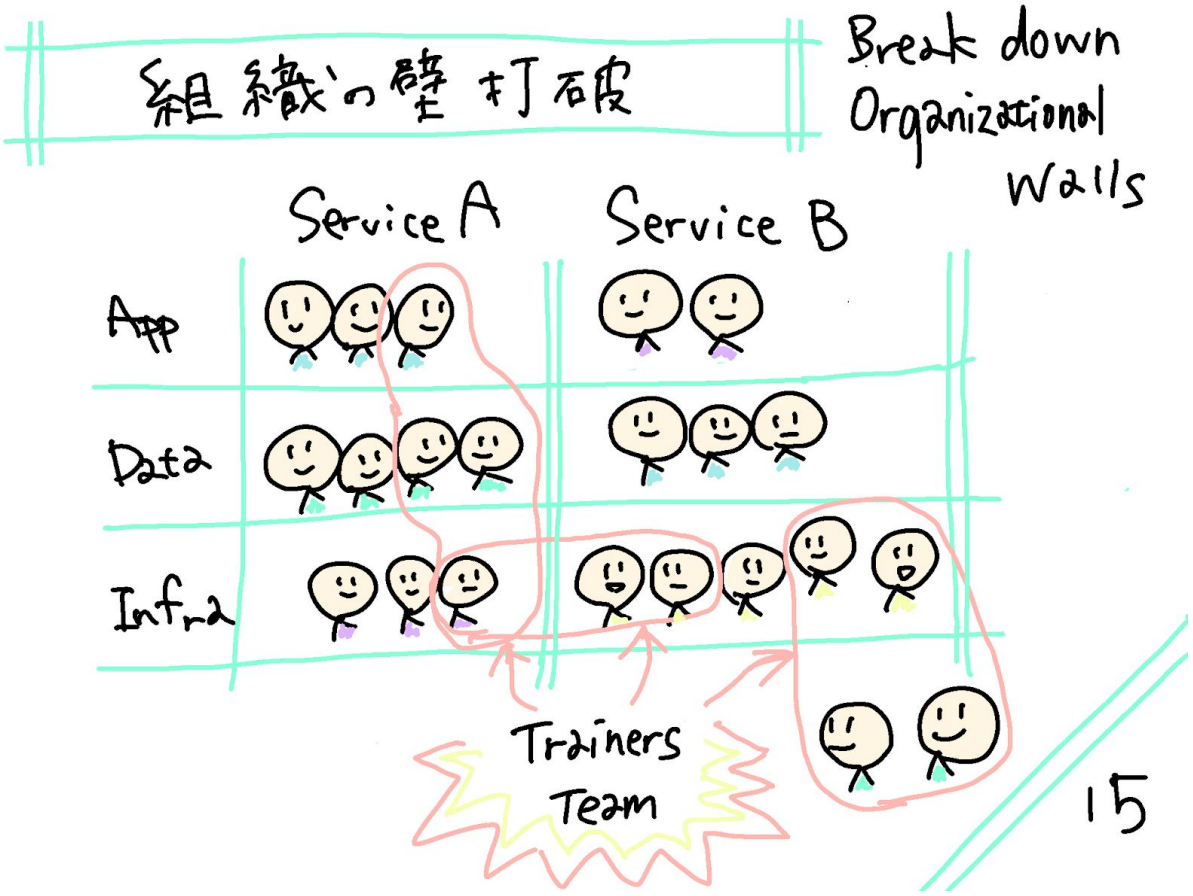
　つらい。１人だと仲間と励まし合うことも なく、忙しいと精神的負担も増える。

4. 解決方法

　各研修において、トレーナーをチーム体制にする。複数名で研修開発をすると、タスクを分担でき、責任感も生まれる。辛いときはお互いに励まし合いながら、目標を持って、明るく開発を進められる。

5. 結果状況

　複数人でタスクを分担し責任感を持って開発していると、ひとりぼっちで研修を開発するよりも短い期間で完成することができる。




組織の壁打破<15>

1. 状況

　大きな組織では部署やレイヤーごとにアーキテクチャーも異なり、組織を越えてエンジニア同士のコミュニケーションはなされないことも多い。

2. 問題

　１チームで起こっている新しいノウハウやベストプラクティスが他のチームに展開されづらい。

3. フォース

　自分から積極的に他のチームと絡もうとするエンジニアが少ない。

4. 解決方法

　プログラム責任者は、部署やレイヤーの壁を越えて、トレーナーチームを作る。研修担当人事は、異なるトレーナーチーム同士が情報交換できる場を設ける。

5. 結果状況

　研修開発を通し、エンジニア同士が組織の壁を越えてコミュニケーションをとるようになる。トレーナーチームに複数の部署のメンバーがいると、研修内容が偏らなくなる。今まで関わりのなかった人たちがノウハウを共有し合い、より良い研修ができるし、トレーナーたち自身も新しい事例を知れる。通常業務についても情報交換し合う関係性が、組織を越えてできる。


専門家のアドバイス<16>

1. 状況

　社内のエンジニアが講師をやる場合、指導経験がないことがある。

2. 問題

　社内の状況に合った研修を開発できるが、一般的なコンテンツに自信がなかったり、研修の作り方や教え方に自信がない場合がある。

3. フォース

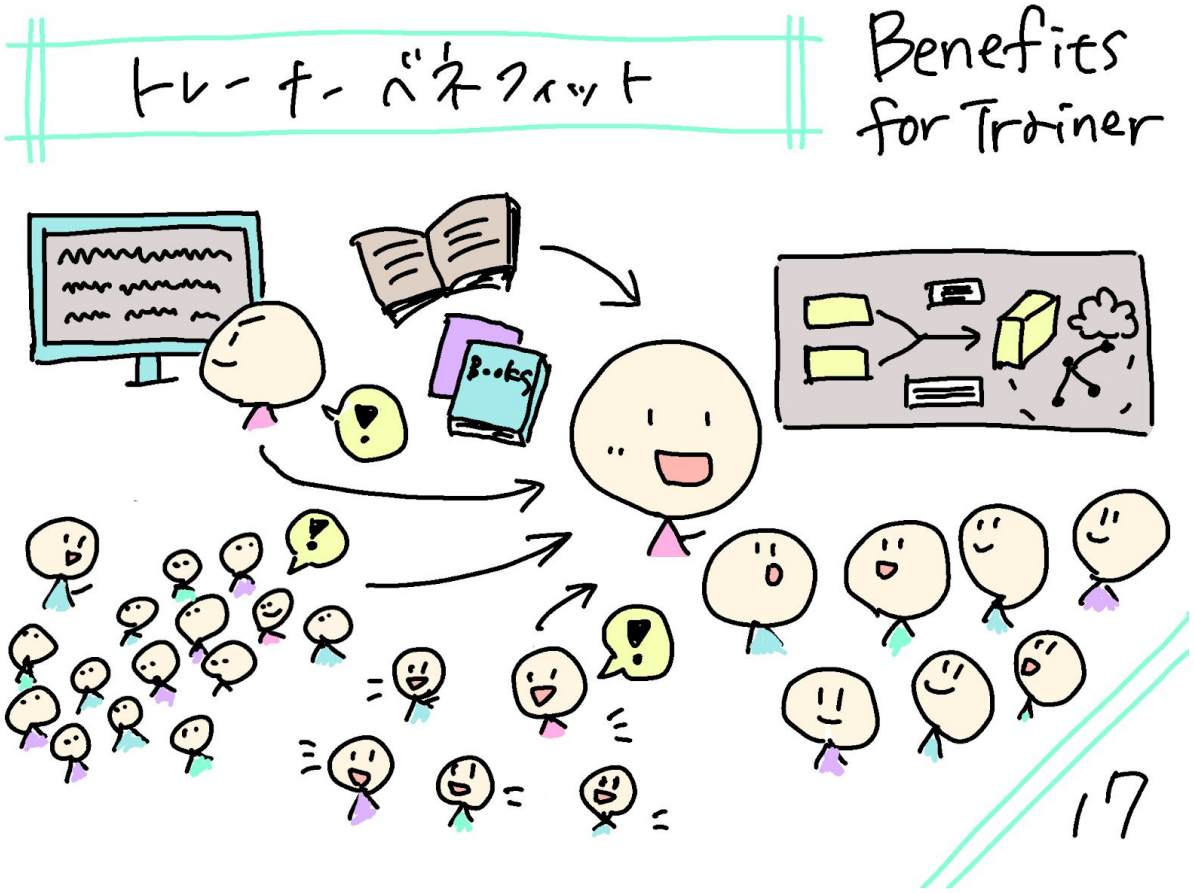
　トレーナーという立場だと他の人の意見を求めにくかったり、自信がないことを表に出しづらい。同じ立場や自分より専門的な立場の人がいれば、相談しやすい。

4. 解決方法

　研修担当人事が、社外のその分野における専門家に強力を依頼する。社外の専門家とトレーナーが話せる場を設けたり、必要であれば一緒に研修開発をしてもらえるよう調整する。

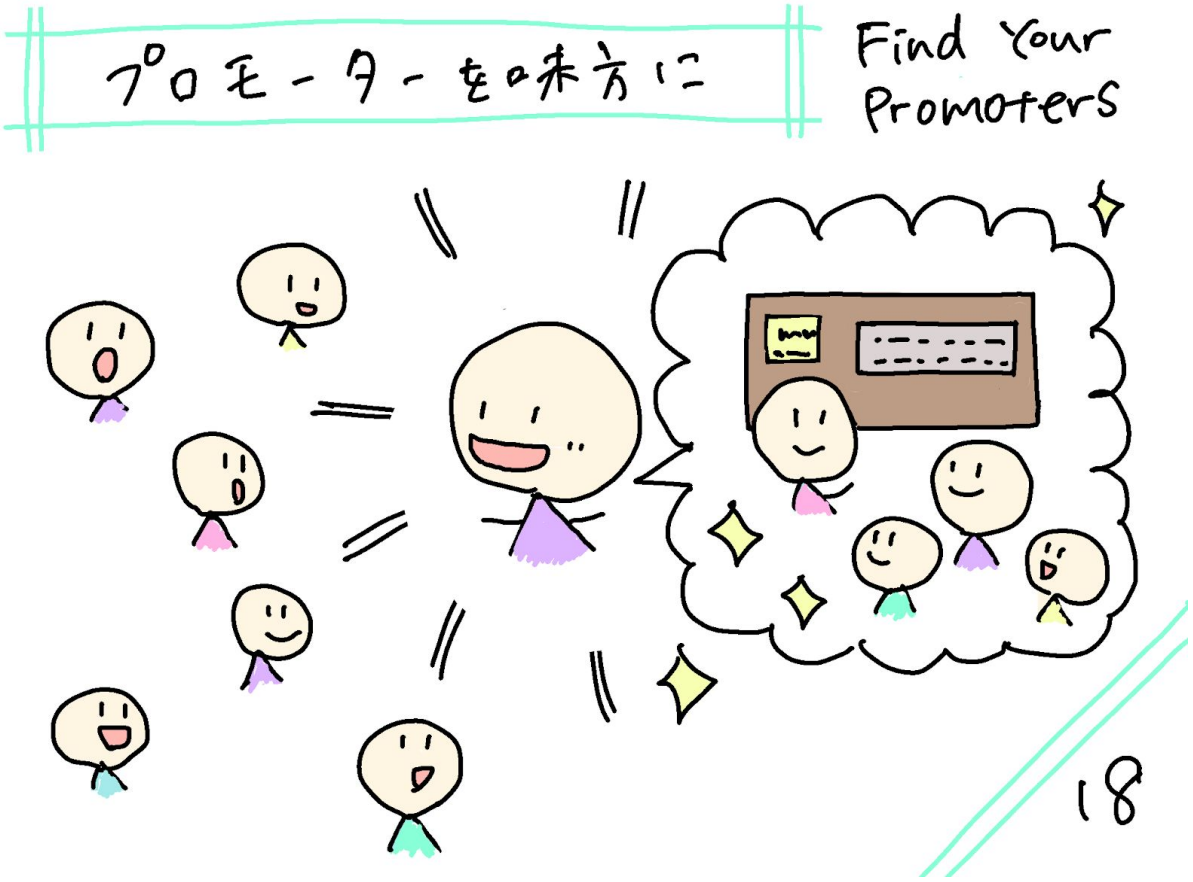
5. 結果状況

　社内のエンジニアがコンテンツを考えるため、社内の事例やニーズにも対応しているが、社外の専門家のアドバイスも組み込まれているため、その分野で必要な知識と社外の事例も含まれた研修が完成する。トレーナーにとっては、社外の人との交流が専門的なスキルの勉強になり、成長の機会にもなる。




トレーナーベネフィット<17>

1. 状況
　トレーナーになるエンジニアは、通常業務も忙しい中、研修の開発と実施に協力する。研修担当人事は、研修開発を評価に加えるようトレーナーの上司に依頼もする。しかし、それだけでは研修開発のやる気には繋がらないエンジニアもいる。
2. 問題
　エンジニアはやりがいを感じなければ、研修開発に積極的に取り組まなくなる。
3.フォース
　技術を教えることやベストプラクティスを広めることなどにやりがいを感じるだけでなく、わかりやすく受け取れるベネフィットもあれば嬉しい。
4. 解決方法
　研修担当人事は、トレーナーに対して、研修やカンファレンスの費用面でのサポートをし、トレーナーだけが集まる情報交換の場、専門家のアドバイス<16>をもらえる場などをできる限りアレンジする。トレーナーランチや飲み会なども開催し、研修開発が楽しいと思ってもらえるように心がける。
5. 結果状況
　「トレーナーをやらなければ経験できなかったことが経験できた」と思ってもらえれば、やりがいを感じてもらえる。トレーナーのための場のアレンジにより、トレーナー同士の繋がりも深まり、情報交換が活発に行われるようになる。トレーナーになりたい人が増える。




プロモーターを味方に<18>

1. 状況
　研修を開発しても、ただ実施するだけでは認知度は上がらない。
2. 問題
　研修担当人事だけの力では、社内での認知度をあげるのは難しい。
3. フォース
　人やチームによって好まれる情報収集手段が違うので、全体メールや社内掲示板などの社内でお決まりの案内方法では届かない層がいる。
4. 解決方法
　あらゆるSNSツールを駆使して宣伝をしたり、エンジニア向けのイベントを開催したりして、研修の認知度をあげる。特に、アーリーアダプターと呼ばれる新しいもの好きな人たちを積極的に巻き込み、研修の宣伝をするプロモーターになってもらう。
5. 結果状況
　SNSを見た人やイベントに参加した人、アーリーアダプターたちが拡散源となり、認知度が上がり、研修の受講希望者や研修を応援してくれる支持者が増える。研修開発チームは声をかけられたり、相談を持ち掛けられたりする機会が増え、それを実感する。

## 5. 結論と今後の課題

このように、本論文における研修担当人事のような専門知識がない人間でも、その分野の人を複数人巻き込むことによって、ある分野の研修を迅速に複数開発することができる。また、アジャイルの考え方やフレームワークを取り入れて開発することにより、改善を繰り返してより良い研修を受講者に提供し続ける仕組みもできる。

しかし、研修を頻度高く継続していくためにはトレーナーたちの日々の業務の時間を削り、研修の開発と実施に時間を費やしてもらう必要がある。イノベーターである彼らの時間の多くを研修実施にかけるのは難しく、新しいトレーナーを育てる仕組みを作り、現場の負担になりすぎないようにするのが今後の課題である。

参考文献:


1. Manns, M. L. & Rising, L. (2004). *Fearless Change: Patterns for Introducing New Ideas*. Addison Wesley Professional. (邦訳: 川口恭伸監訳『Fearless Change アジャイルに効く アイデアを組織に広めるための48のパターン』, 丸善出版, 2014年)

2. Schwaber, K. & Sutherland, J. (2017). *The Scrum Guide*. (邦訳: 角征典訳『スクラムガイド』)