実験的アプローチによる現場改善

三菱電機株式会社
細谷 泰夫

改善　＝　変化

改善するときにやること。

- 「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」であると言える。

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）撤退の選択をしずらくなる。

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: 「改善のスコープ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　改善のスコープをきわめて小さくする

- 改善のスコープが大きいと、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。

改善のスコープ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「改善のスコープ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。

実験的アプローチの6つの活動

- 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。

課題と手段のマッピング

- 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。

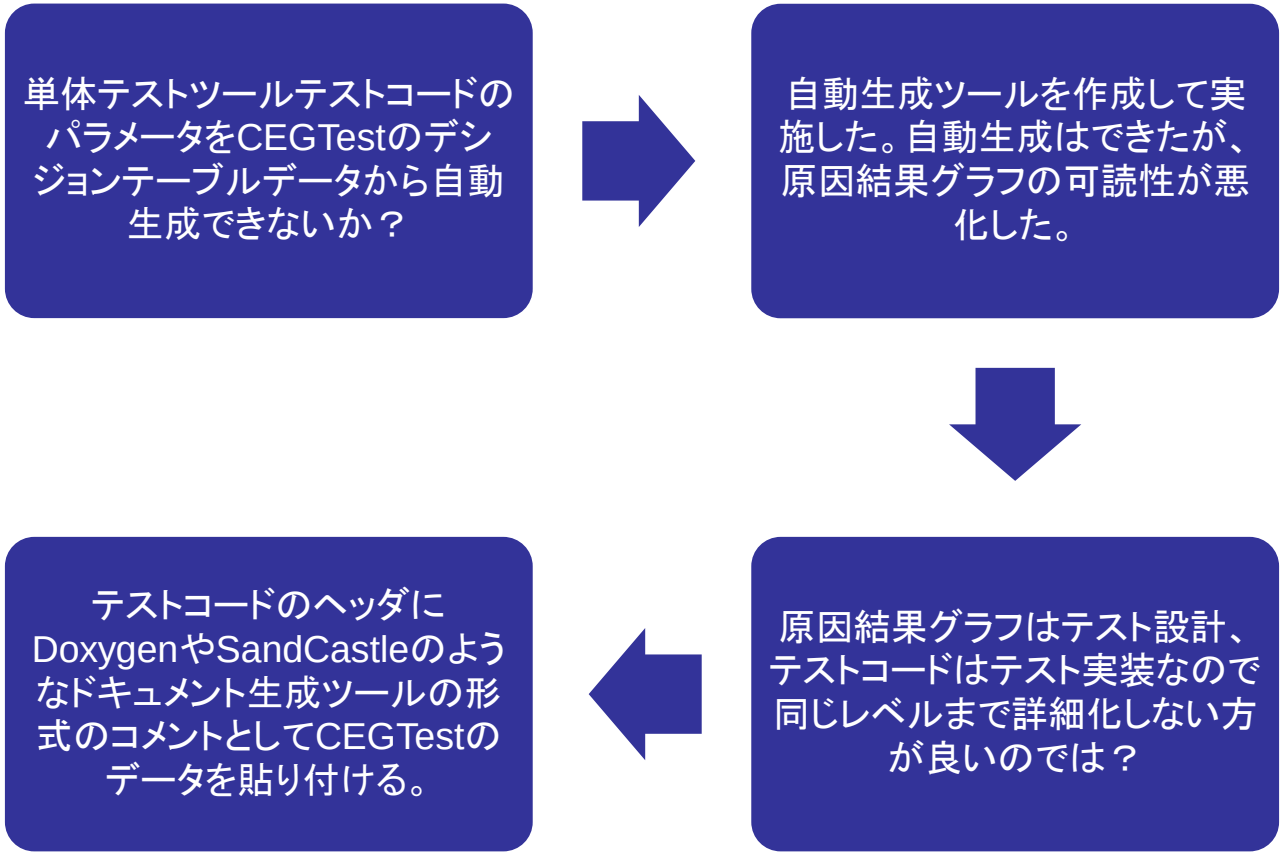
結果の評価

- 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。

事例1 原因結果グラフを単体テストに活用

活動	実施内容
情報収集	JaSSTでCEGTestを知る。ブログ等で内容の確認。
課題の発見	方式設計～単体テストを繰り返すプロセスの中で、単体テストケースを入出力値のパラメータ表で設計していたがテストケースの意図が判りにくいという課題が出た。
課題と手段のマッチング	テストケースの意図を表現する手段として原因結果グラフがマッチング。
手段の試行	CEGTestを利用する上での準備はほとんど必要なかったので試行は省略
手段の実践	CEGTestを使って単体テスト設計を実施。原因結果グラフをレビュー。CEGTestのデータはエクスポートしてソースとは別管理。
結果の評価	テストコードとCEGTestのデータが別管理なので煩雑。元のパラメータ表よりCEGTestの方が意図が読み取りやすい。

事例1の変遷



反復の中で試行を繰り返す。

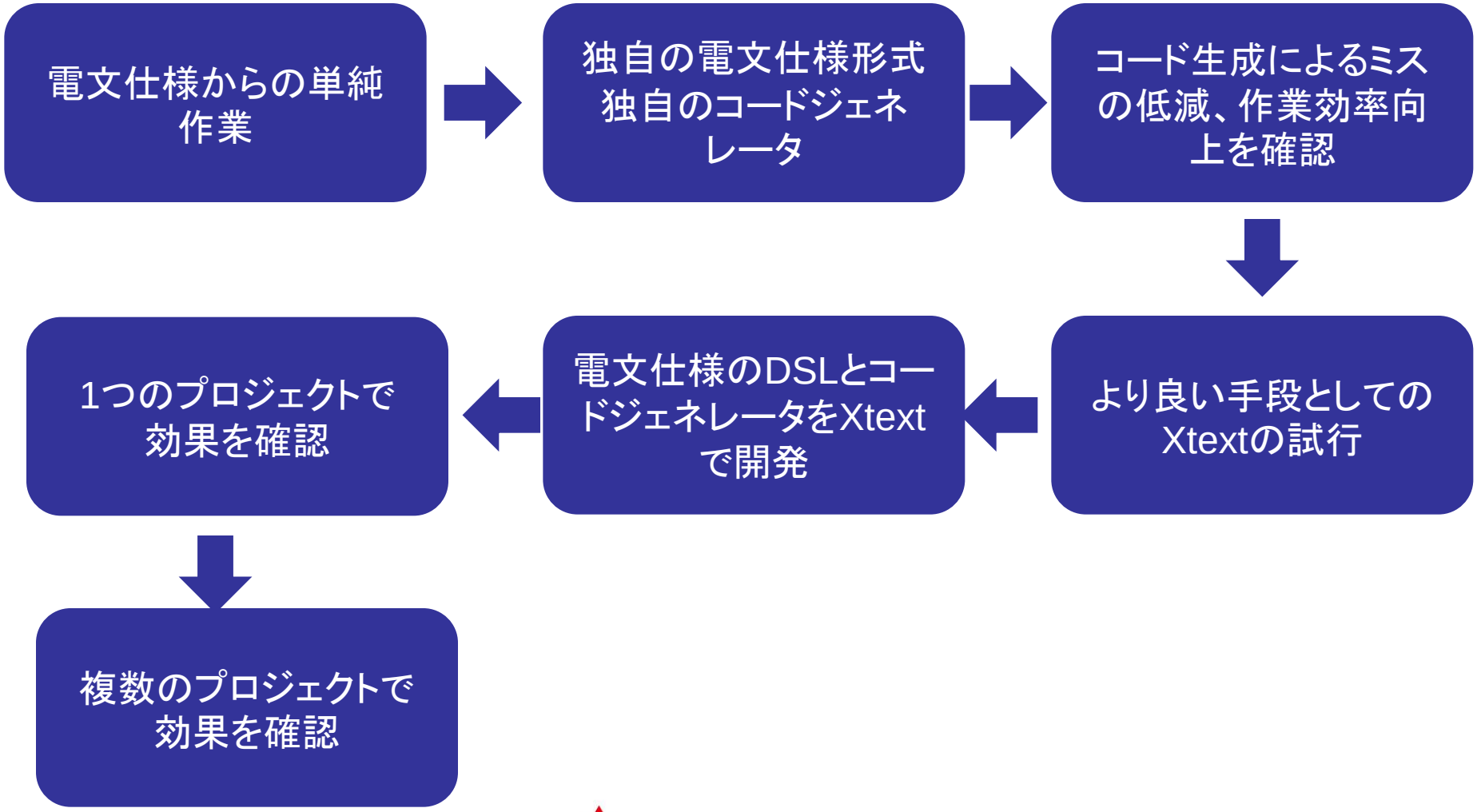
事例1の変遷後の利点

- デシジョンテーブルとCEGTestのデータがテストコードと同じ箇所に一体となっているため、テストコードで実装しているテストの意図が読み取りやすい。
- ドキュメント生成ツールで単体テスト仕様を出力することができる。
- テスト駆動開発と原因結果グラフによるテスト設計を併用することで、コードの内部品質が向上した。
- ペアプログラミング時にペアで原因結果グラフを作成することによって、ナビゲートの地図を得ることができ、ペアプログラミングがスムーズに実施できるようになった。

事例2 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。チュートリアルの試行。
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2の変遷



事例2の変遷

2012.11
電文仕様からの単純作業

2012.12
独自の電文仕様形式
独自のコードジェネレータ

2012.12
コード生成によるミスの低減、作業効率向上を確認

2012.12
より良い手段としてのXtextの試行

2013.1
電文仕様のDSLとコードジェネレータをXtextで開発

2013.1
1つのプロジェクトで効果を確認

2013.2～5
複数のプロジェクトで効果を確認

三菱電機株式会社

事例2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）

2010年～2012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

(3)ツール

TFS、Test Manager

Sphinx

Xtext

実践を通じてコンテキストに適合させる

- 誰かのうまくいった事例の感想として、「たまたまうまくいった」と言われることがあるが、そのコンテキストに適合した方法になっているからうまくいっている。しかし、その必然性を認知しているとは限らない。必然性を認知して言語化するには、コンテキストと実施内容との関係を認識する必要がある。
- 実験的アプローチは、実践を通じて、その効果を経験し、コンテキストとのすり合せを繰り返しながら、そのコンテキストに対して必然性のある改善を進めていく手法と言える。