

SPI Japan2016 発表概要集

・セッション順番 > 発表順番に記載しています。

1A1 「Yahoo! JAPAN におけるアジャイル開発の普及戦略」 山口鉄平(ヤフー).....	2
1A2 「Agile を活用したワーキンググループ活動」 服部悦子(住友電工情報システム)	4
1A3 「現場から始めるアジャイルの技術プラクティス」 岡本卓也 (富士通).....	9
1A4 「『ワクワクする未来を創造する』 アジャイルな組織を目指すツボ」 陸野礼子(日新システムズ)	13
1B1 「プロセス改善活動をスムーズに立ち上げるための取り組み<その2>」 中村英恵(NTT データ).....	18
1B2 「プロセス分析に基づくドキュメント再構成によるプロセス改善」 小島裕次(デンソー).....	24
1B3 「プロダクトライン開発におけるプロセスのコア資産フィードバックモデルの提案」 林健吾(デンソー)	27
1B4 「ビデオ撮影及び画面キャプチャによる分析プロセスの改善」 原田大輔(住友電工情報システム株式会社)	30
1C1 「開発者は不具合をどのように捉えているのか？不具合票のテキストマイニングをはじめる」 小島義也(エプソンアヴァンシス)	35
1C2 「XDDP 導入／定着の失敗事例に対する病理学的処方箋」 葛西孝弘(日本電気通信システム)	44
1C3 「なぜなぜ分析はこれでうまくいく！ ~根本原因特定へのアプローチ~」 林真也(住友電工情報システム)	48
1C4 「ソフトウェア開発における品質管理の理論と実践」 小室睦(プロセス分析ラボ).....	55
2A1 「マンスリー自己申告とクロスチェックの高速 SQA で赤字案件を削減」 木戸寛(大和コンピューター)	59
2A2 「失敗に学ぶ。再発防止策の立案・実行から定着するまでの仕組み作り」 井之内博夫(オムロンソフトウェア).....	63
2A3 「危険予知トレーニングによる障害の未然防止」 南波貴紀(インテック)	67
2A4 「開発チームと QA がフィードバックし合いながら成長するアジャイル開発の改善事例」 永田敦(ソニー)	71
2B1 「全員参加型画面レビューのススメ」 長谷川真裕(インテック)	80
2B2 「作りすぎない基幹システム刷新プロジェクト」 伊藤茂泰(富士通コワーコ).....	85
2B3 「TOC に基づく日立グループ 1,000 人に広がる業務改善 6 ヶ月プログラム」 ハ木将計(日立製作所)	89
2B4 「永遠の目標「ヒューマンエラー ゼロ」達成の秘訣」 渡辺聰美(富士通エフ・アイ・ピー)	104
2C1 「複数部門にまたがったシステム・ソフトウェア開発プロセス改善の取り組み」 藤山晃治(パナソニック)	109
2C2 「通信機器組込み用ソフトウェアの開発プロセス改善による品質向上の一実施例」 嘉屋良洋(アンリツネットワークス).....	112
2C3 「サービス提供組織でのプロセス評価・改善の試み」 大瀧陽悦(NEC ソリューションイノベータ)	117
2C4 「ISO26262「確証方策」のより効率的な実施の取り組み」 菅沼由美子(パナソニック)	121
3A1 「ふりかえりを用いたレビュー方法の改善」 木村慎吾(インテック)	125
3A2 「やらされ感の壁を越えて、現場と一緒に改善を考えるための第一歩」 江崎美保(日新システムズ)	128
3A3 「「対話」を加えた KPT ふりかえりによる、チームの活性化」 長岡桃子(富士通).....	132
3B1 「品質コックピット構築による品質改善」 松井伸一(NEC ソリューションイノベータ)	137
3B2 「社会インフラシステムにおけるプロジェクトの異常予測手法の適用」 飯村拓志(東芝).....	140
3B3 「品質特性に基づく品質メトリクスの定義と活用」 相澤武(インテック)	144
3C1 「仮想化技術を使って、みんなでチャレンジ！」 佐藤直哉(東芝テック)	149
3C2 「「あるある診断ツール」による保守/運用課題の見える化」 室谷隆(TIS)	152
3C3 「信頼度成長曲線の導入による統合テストの改善」 中村伸裕(住友電工情報システム)	156

1A1 「Yahoo! JAPAN におけるアジャイル開発の普及戦略」 山口鉄平(ヤフー)

<タイトル>

Yahoo! JAPAN におけるアジャイル開発の普及戦略

<サブタイトル>

<発表者>

氏名（ふりがな）： 山口 鉄平（やまぐち てっぺい）

所属：ヤフー株式会社

<共同執筆者>

氏名（ふりがな）：

所属：

<要旨>

現在、Yahoo! JAPAN では 50%程度のサービスをアジャイル開発により開発している。本発表では、Yahoo! JAPAN の約 2000 名の多人数かつ多チームのモノづくりに関わる組織において、どのようにアジャイル開発を普及させたかを紹介する。特に、複数の普及施策を普及のフェーズごとに分け、それぞれの施策自体とその根拠と目的、施策結果などを紹介する。

<キーワード>

アジャイル、普及施策、普及のフェーズ、サービス開発

<想定する聴衆>

SEPG 初心者、ソフトウェアエンジニア、ソフトウェア開発のマネージャ

<状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

全社のサービス開発の支援チームが社内のサービス開発の改善に向け、社内でのアジャイル開発の普及に取り組んだ。

2.改善前の状態

改善前の社内状態としては、下記 3 点の課題が発生していた。

- ビジネス部門と開発部門の業務目標が異なっていた。
- 両部門の窓口担当者がボトルネックになり情報の速度と精度が低下していた。
- 部門の足並みをそろえるための調整コストが大きかった。

3.改善前の状態をもたらした原因（因果関係）

各サービスのビジネス部門と開発部門が一緒になってサービス開発に取り組んでおらず、社内受発注による弊害が発生していた。

4.計画した変更内容

様々なサービスが、サービスごとのビジネス部門と開発部門が一緒になって、かつ素早くかつムダの少ないサービス開発を実現するために、アジャイル開発によるサービス開発の全社的な実施を目指した。

5.変更の実現方法

普及のフェーズごとに、累積的な普及施策と順次的な普及施策を実施し普及をおこなっている。なお、現在は普及期に該当する。

- 支援初期では、初期採用層を捕らえ支援を行うとともに。その支援の際には採用のハードルを下げる工夫をおこなった。加えて、支援を行う際には失敗確率を下げる工夫をおこなった。
- 普及期前では、普及率向上に向け、事例紹介による実績の横展開やインターネットでの情報展開、社外での発表を社内で伝えることなどをおこなった。
- 普及期では、支援対象の規模が拡大するための対応として、全社の教育システムに組み込むことや、影響力を持つメンバーのネットワーク化などをおこなっている。規模の拡大に伴い、支援チームによる支援先への関わりは減少させている。
- 予想される成熟期では、支援チームとしての活動は終了し、影響力を持つメンバーが独立して活動することやそれぞれの交流によるノウハウ共有・手法の発展を目指す。

6.変更後の状態や改善効果

上記施策をおこなった後、社内で開発プロセスに関するアンケートをおこなった。それにより、アジャイル開発によるサービス開発が社内での 50%となり、当初の課題を解決し、チームから提供する機能がおおむね向上したことがわかった。

7.改善活動の妥当性確認

支援チームによる課題解決は進んだが、まだ下記の課題が存在する。

- アジャイル開発の普及は進んでいるものの、まだ道半ばであり、今後訪れる成熟期で発生する課題がわかっていない。
- チームごとに、アジャイル開発の利用の質が異なり、我々から見ると質の低いアジャイル開発をおこなっているようなチームが存在する。

<タイトル>

Agile を活用したワーキンググループ活動

<サブタイトル>

<発表者>

氏名 (ふりがな) : 服部 悅子 (はっとり えつこ)

所属 : 住友電工情報システム株式会社 QCD 改善推進部品質改善推進グループ

<共同執筆者>

氏名 (ふりがな) :

所属 :

<要旨・主張したい点>

弊社ではワーキンググループ活動は活発に行われているが、若手メンバー中心のあるワーキンググループで活動が停滞する事態が発生した。そこで活動を活発化させることを狙い、Agile のプラクティスをワーキンググループ活動に取り入れることを実践した。結果、Agile はワーキンググループのような改善活動にも有効であることが確認できたので、その事例を紹介する。

<キーワード>

改善活動、ワーキンググループ、Agile、Scrum、改善推進者、若手メンバー

<想定する聴衆>

プロセス改善推進者、プロジェクトマネージャ・SE・プログラマーなど改善活動参加者

<活動時期>

2015/4～2016/3(活動継続中)

<活動状況>

- 着想の段階 (アイデア・構想の発表)
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他 ()

<発表内容>

1.背景

当組織では毎年、品質と生産性に関する事業目標を設定し、目標達成に必要なワーキンググループ(WG)を複数立ち上げ活動している。ここ数年、WG活動は非常に活発で開発部門メンバーの半数以上がいずれかのWGに属し改善活動に参加している。

その中のプログラム開発WGは若手メンバー(開発経験年数が5年程度)に参加してもらい、UT設計時間の短縮を目標に改善策の検討を進めていた。最初は活発に活動できていたが、時間が経つにつれメンバーの出席率が下がり会議が中止になることも増え、結果として改善活動のアウトプットが出ない状況になっていた。

参加メンバーは本人とその上長に16MH/月をWG活動に割り当てる事を了承してもらっていた。しかし、実際の活動工数を確認したところ、WG(会議)参加時間、個人ワーク時間ともに予定工数より少なく、さらに図1に示すように3ヶ月周期で活動時間が減っていくという状態であった。

表1 参加メンバー1人・1ヶ月あたりの活動工数

活動工数	予定工数(一ヶ月あたり)	実績工数(一ヶ月あたり)
WG(会議)参加時間	8MH(1回2H×4回)	1~6MH
個人ワーク時間	8MH(毎週2MH)	1~2MH

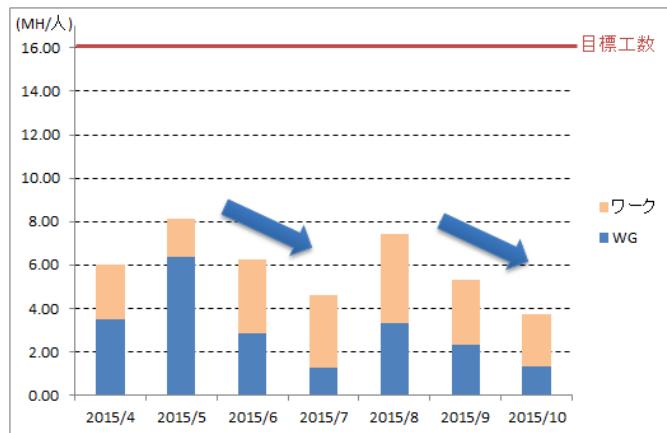


図1 WG活動に参加する一人あたりの投入工数

2.改善したいこと

以下3点を改善したいと考えた。

- 改善活動への投入工数を増やす
- 改善活動に対して一定の投入が継続して行われる
- 改善活動の成果を出す

3.改善策を導き出した経緯

従来、改善活動はおおよそやるべきことを決めてからWBSを作成し、各個人にタスクを割り当て宿題形式で進めていた。活動を振り返ると以下3点の問題があることがわかった。

- 問題1. レビューで指摘が多く修正とレビューを繰り返し、完了までに時間がかかる
- 問題2. 宿題をやってこない(もしくはWGに欠席する)ので完了までに時間がかかる
- 問題3. 状況が進むにつれ新たにやるべきことが発生し、当初予定していたものが完成するまでに時間がかかる

私は過去に開発プロジェクトで Agile(Scrum)を試行した経験があった[1]。Agile はイテレーションを繰り返すことによりプロジェクトで起こる変化に適応していくことが特徴であり、これが問題 3 を解決するのではないかと考えた。また[1]より開発者のモチベーション向上や若手メンバーの能力向上に効果があることがわかつていたので問題 1, 2 も効果がありそうに感じた。よって Agile のプラクティスを WG 活動に応用し、これらの問題を解決することにした。

4.改善策の内容

Agile のプラクティスを参考に以下 3 つの対策を考えた。

対策 1. イテレーションの導入

Scrum のスプリントのように短い期間でゴールを設定しておくことにより、新たに見つかることや課題をバックログに切り出して目前のタスクに集中する（問題 3 の解消）。

対策 2. プロダクトバックログ

対策 1 を実施するために WG の目標を分割してプロダクトバックログにし、そのスプリントで行うバックログを自分達で選択する。自分が興味あるものを選択するので質が良くなる（問題 1 の解消）。タスクの実施手順がイメージできるものを選択するので確実に実施される（問題 2 の解消）。

対策 3. ペアプログラミング

1 つのバックログを一人ではなくペアプログラミングの様に 2 名ペアで取り組む。2 名で相談するので質が良くなる（問題 1 の解消）。2 名で取り組むので確実に実施される（問題 2 の解消）。ペアの片方が WG を欠席しても、もう一方が状況を説明できる（問題 2 の解消）。

5.改善策の実現方法

(1) 教育

スクラムガイド[2]を WG メンバーに説明した(15 分程度)。その他のトレーニングは行っていない。

(2) 導入

3 つの対策を実現するため、Scrum をベースに以下のように活動することにした。

・プロダクトオーナー

改善推進者の立場で参加している私がプロダクトオーナーとなった。

・スプリント(イテレーション)

WG は毎週 1 回 2H の会議体で行っており、4 回(1 ヶ月)を 1 スプリントとした。(図 2)

・プロダクトバックログ

WG の目標を分解し、リストアップしておいた。

・スプリントゴール

今月完成させる成果物（もしくはタスク）をプロダクトバックログの中からメンバー自身で決めた

・ペアプログラミング(ミニチーム)

2 名の場合、どちらか一方が欠けると結局一人になってしまふので 3 名のミニチームにすることにした。

この WG は 9 名のメンバーがいたのでミニチームを 3 つ編成した。

チームメンバーは固定にせず、毎月各人がやりたいと思うバックログを優先して編成した。

※注:ペアプログラミングは Agile の XP のプラクティス

・スプリントバックログ

毎月 1 回目の WG で、ゴールに決めた成果物を完成させるために必要な作業をミニチームで相談して決めた。

・デイリースクラム

毎月 2、3 回目は成果物の中間レビューにあてた。

また WG 以外にも、ミニチームでの打合せが自主的に行われた。

・スプリントレビュー

毎月の最終回を成果物レビューの回にし、できあがったものをメンバー全員でレビューし完成できたかどうか全員で判断した。

・レトロスペクティブ

毎月の最終回で今月の活動内容について KPT 方式(*)で振り返りを行った。

(*)Keep、Problem、Try の略

・バーンダウンチャート

今回は利用しなかった。



図 2 1ヶ月の活動 (1スプリント)

6. 改善による変化や効果

(1) 投入工数の増加

一人あたりの活動工数を確認したところ、会議への参加時間は微増であったが会議以外のワーク工数が対策後は 2.6 倍になった（図 3）。これはミニチームでの議論が活発にされていることを示している。1 ヶ月あたりでみると、期待している 16MH/月に近い工数を投入できている。

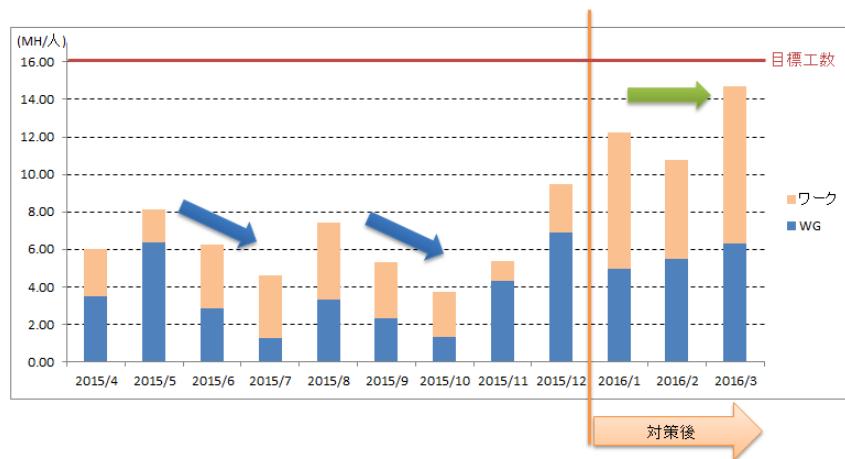


図 4 WG 活動に参加する一人あたりの投入工数(対策後)

(2) 活動の継続

対策前は 3 ヶ月周期で投入工数が減少していく傾向が見られたが、図 5 の通り対策後はその傾向は見られず継続して一定量の活動が行われている。1 ヶ月単位のスプリントにより「ゴール」が見える化されることが活動の継続に繋がっていると考える。またペアプロ的なミニチームによる学習効果やチームメンバーが共同で進めているということも貢献している。

(3) 活動成果(アウトプット)の増加

3 ヶ月間の活動で 9 件のバックログが完成した。以前のやり方では一人で担当していたので完成までに 2 ヶ月以上かかっていたものが 1 ヶ月で完成するようになった。

7.改善活動の妥当性確認

(1) 施策の妥当性

3つの問題点が解消できた。

問題1. レビューで指摘が多く修正とレビューを繰り返し、完了までに時間がかかる

→ミニチームとして複数のメンバーで取り組む事により、一人の思い込みで進めてしまうことが予防され、レビューでやり直しになることが減った。また月末までの完成に向けメンバー内で分担して作業を進めることができた。

問題2. 宿題をやってこない（もしくはWGに欠席する）ので完了までに時間がかかる

→欠席者がいても出席しているメンバーが報告し、全体の進行を妨げることは減った（あるミニチームの全員が欠席になったのは12回中1回であった）。

問題3. 状況が進むにつれ新たにやるべきことが発生し、当初予定したものができるまでに時間がかかる

→レビュー中に見つかった課題は別のバックログに切り出すことを提案し、各月のゴールに向けて集中して作業できた。

(2) 悪影響の確認

改善活動は開発部門からすると間接業務であり、間接比率が上がりすぎることは望ましくはない。今回の結果では期待している範囲内（16MH/月）で従事し、予定の間接比率に収まっている。またメンバーは開発業務を抱えながら改善活動に参加するので過剰な投人が起こることは少ないと考えている。

(3) 今後の課題

1ヶ月で終わらないタスクも2件あった。例えば「業務フローを作成する」というバックログについて担当者の経験不足からシステムフローになってしまふようなことがあり、1スプリントで完成できなかった。若手メンバーの経験不足や知識不足によってリワークが発生するものは防げない。ミニチームの中にある程度経験のある先輩社員を入れるなどチーム編成の工夫も必要である。

A. 参考文献

[1] SPI Japan 2012 企業セッション カイゼン力の育て方（住友電気工業）「アジャイルの試行」

http://www.jaspic.org/event/2012/SPIJapan/session0C/0C1_ID004.pdf

[2] アジャイルガイド

<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-JA.pdf>

1A3 「現場から始めるアジャイルの技術プラクティス」 岡本卓也 (富士通)

<タイトル>

現場から始めるアジャイルの技術プラクティス

<サブタイトル>

～ユニットテストから勝手に始めてみよう～

<発表者>

氏名 (ふりがな) : 岡本 卓也 (おかもと たくや)

所属 : 富士通株式会社

<共同執筆者>

氏名 (ふりがな) :

所属 :

<要旨・主張したい点>

アジャイル型開発に挑戦する動きが拡がっているが、様々な課題に直面し苦戦している事例が散見される。本事例では、課題の中でも現場レベルで解決に向けた活動が比較的容易と思われる技術的なプラクティスに着目し、その中でもキープラクティスとなるユニットテストを導入した。ここで行った取り組みとその結果について報告する。

<キーワード>

アジャイル、技術、プラクティス、現場、ボトムアップ、ユニットテスト、プロジェクトファシリテーション

<想定する聴衆>

ソフトウェアエンジニア、アジャイルに興味のある方、アジャイル導入に苦戦している方

<活動時期>

2014年7月～継続中

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他 ()

<発表内容>

1.背景

当社ではキャリア向け光伝送装置の制御用ソフトウェアを開発している。近年の要求では大規模/高機能化かつ短納期化が顕著になっており、これに応えるため開発効率の向上を目的として従来のレガシー開発スタイル(ウォーターフォール型開発)からアジャイル型開発スタイルへの変革に取り組んでいるが、様々な課題により導入が進んでいない。

課題の一例

- 1) 組織やプロセスの課題
- 2) 技術の課題

1) の組織やプロセスの課題については、顧客との契約形態や他部門との連携/作業標準など、組織的かつトップダウン的に取り組む必要があり現場のマネージャレベルで挑戦できる事は限定的である。それゆえに、この部分に壁を感じてアジャイルへの取り組みに苦労しているケースも多い。

一方で、2) の技術の課題に関しては、現場で挑戦できる領域が多く、ボトムアップ的に取り組むことが可能である。

本発表では、上記の 2) 技術の問題に取り組んだ活動事例について報告する。

2.改善したいこと

現場の技術レベルが低く、必要なアジャイルプラクティスを実践できない。

アジャイルプラクティスに必要な技術スキルは多岐にわたるが、どこから手を着けて良いか分からず。

3.改善策を導き出した経緯

各種の技術的アジャイルプラクティスを調査し、そこで必要となる要素技術や他のプラクティスとの因果関係/依存関係を調べた。その結果、多くのアジャイルプラクティス実践のためにはユニットテスト(自動化された単体テスト)が重要なキーであり、これを実施するための前提条件(オブジェクト指向や自動化のための環境)や、実施することによる波及効果(リファクタリング可能、設計品質の向上)など、多くのアジャイルプラクティスに影響を与えることが分かった。

4.改善策の内容

従来は手動で行っていたモジュール単体テストを止め、代わりにユニットテストを導入した。

ユニットテストを実施するための一連の周辺技術と環境を導入した。

具体的には下記の活動を行った。

メンバのトレーニング

- ・ オブジェクト指向プログラミングの基礎教育(勉強会/読書)を実施した。
- ・ コードレビューを徹底し、設計/コーディングスキルの向上を図った。

テストフレームワークの導入

- ・ プロダクトコードに Google Test フレームワークを導入し、従来のモジュール単体テストにおける試験項目書/試験手順書の代わりにテストコードを書いた。
- ・ 作業工程として、従来の手動で行っていたモジュール単体テストをユニットテストで代替可能とした。

CI によるメトリクス測定と結果の可視化

- ・ CI によるユニットテスト起動と結果のメトリクス集計/可視化を行い、ユニットテストが常に PASS 出来ている状態を維持した。

上記を実施するための環境構築

- ・ 版数管理(VCS) : Mercurial/Git
- ・ チケット管理(ITS) : Redmine
- ・ CI 環境 : Jenkins
- ・ コードレビュー環境 : Reviewboard

5.改善策の実現方法

工夫した点

- ・ メンバが新しい方法に前向きに取り組めるようにプロジェクトファシリテーションに気を使った
朝ミやカンバンでチームの作業リズムを作り、振り返りで継続的な改善を行った。
チーム内のフランクなコミュニケーション促進に注力し、「分からぬこと」を気軽に質問したり、教え合ったりすることに対する心理的なハードルを下げた。
- ・ 対外的な報告としては、既存の作業プロセスから逸脱していない点の説明を十分に行つた
「アジャイルだから」という特別な扱いをすることによる付帯作業の増大を避けるため、トライする範囲は極力現場マネージャの裁量範囲内とし、チームメンバにはゴールに向かって推進する活動に集中してもらった。

苦労した点とその対策

- ・ 作業負荷の集中とボトルネック化

実際に技術的なトライを開始してみると、保有スキル的にいわゆる「できるメンバ」と「それ以外のメンバ」の差が明確となり、かつ出来るメンバが少数派であった(今回のプロジェクトにおいては、10名中「できるメンバ」は1名のみ)ため、結果的にここが開発のボトルネックとなってしまった。

(対策)

自らの進捗よりも長期的なチームの成長を重視して、開発期間中に勉強時間を多めに確保した。具体的に最も学習効果の高かったのは、設計レビュー/コードレビューを全員参加型で行い、その場で良い設計/良い実装/良いソフトとは?という認識を「できるメンバ」から「それ以外のメンバ」への伝承を徹底させる事であった。対外的にはアウトプットの提供時期が遅くなる(=進捗が遅くなる)ように見えていたが、ここに関しては「上流での設計にリソースをつぎ込み品質を確保する」という説明で乗り切った。結果的には、プロジェクト後半で各メンバのスキルレベル上昇と共にボトルネックが解消され、トータルでは優れた生産性を達成できた。

- ・ 既存の作業標準とのすり合わせ

対外的には特にアジャイル開発を宣言していないこともあり、開発現場としては重要かつ必須な成果物の他にも、既存の作業標準に準拠したドキュメントやエビデンスの作成が必要であり、重複作業による負荷増大が発生した。

(対策)

既存の作業標準に関してはあるレベルで折り合いをつけ重複作業を許容した。これは、根本的解決(作業標準の変更承認)を考えた場合の困難や時間的制限を考え、ある意味トライアル中である今回のプロジェクトにおいては必要工数であるというある意味で消極的な判断となった。ただし、今プロジェクトの成果をベースとして、全体の作業標準最適化への取り組みは今後継続していくべき課題である。

6.改善による変化や効果

導入できたアジャイルプラクティス

- ・ コードの共同所有
- ・ バージョン管理
- ・ 自動ビルド

- ・ 継続的インテグレーション
- ・ シンプル設計
- ・ リファクタリング
- ・ ユニットテスト

直接的効果

- ・ 開発効率向上（他チーム比で約 1.7 倍）
- ・ 品質向上（他チーム比で約 1.5 倍）

その他の波及効果

- ・ 繰り返し開発が可能になり、柔軟な部分リリースが可能になった
- ・ 回帰テストを常に実施できている安心感から、積極的にリファクタリングを行うようになった

7. 改善活動の妥当性確認

活動全体を通して

- ・ 現場からのボトムアップでもアジャイルへのチャレンジが可能なことを実証し、当事者として勇気を持つことが出来た。
- ・ Google Test フレームワークによるユニットテストは、従来手法である手動による単体テストと比較して、網羅性や試験観点の十分性という面で、十分に代替手段となり得ることが分かった。
- ・ 技術的プラクティスを行うためのノウハウや必要環境/ツールなどは、ここ数年で劇的に発展しており、最新の情報を参考することで導入の難易度は大きく下がる。逆に、数年前の常識/技術レベルを前提にしてトライに躊躇しているのであれば、大きな機会損失と考える。
- ・ 事前に「何をどんな風にトライするか」について厳密な計画を立てることは不可能。「常に勉強しチャレンジし続ける」の部分にのみチームとしてコミットし、そこに向けて進み続ける姿勢が重要である。（まさにアジャイルの考え方）

費用対効果

- ・ ツールや環境構築などの直接的なコストはゼロ（全て OSS にて構築）。
- ・ 運用や教育などの間接的なコストは、効率/品質の向上から短期間で回収可能。

残存課題

- ・ まだ活動の属人性が高く、熱意のあるリーダーやスキルのあるエース級エンジニアが不可欠。
- ・ 事例蓄積が不足しており、実績を背景としてのプロセスや組織の変革にまでは手が出せていない。この点を継続して改善していくためにも、トライした活動については可能な限り結果を数値化（メトリクス化）し、実績の積み上げを背景とした議論に繋げていく必要がある。

A. 参考文献

1A4 「『ワクワクする未来を創造する』 アジャイルな組織を目指すツボ」 陸野礼子(日新システムズ)

<タイトル>

『ワクワクする未来を創造する』 アジャイルな組織を目指すツボ

<サブタイトル>

～チームの価値、組織の価値の最大化を考える～

<発表者>

氏名 (ふりがな) : 陸野 礼子 (りくの れいこ)

所属 : 株式会社日新システムズ 未来戦略室

<共同執筆者>

氏名 (ふりがな) : 前川 直也 (まえかわ なおや)

所属 : 株式会社日新システムズ 未来戦略室

<要旨・主張したい点>

ソフトウェア開発でのアジャイルを成功させるには、いかにゴールを意識するかであり、

ゴールを意識しながら、プロジェクトや組織を作っていく戦略を練ることが重要になる。

私たちは、組織のスクラムマスターとなり、アジャイルな組織を目指して活動を開始した。

目的はチームの価値、組織の価値を最大化して、人も組織もさらなる成長をしていくことである。

本発表では具体的な取組みを通して、アジャイルな組織へのアプローチ方法をご紹介する。

<キーワード>

組織改革、風土改革、プロセス改善、人財開発、アジャイル、コミュニケーション、ふりかえり

<想定する聴衆>

組織改革/風土改革に取り組まれている方、SEPG、アジャイルを検討/実践されている方

<活動時期>

2015年6月～現在推進中

<活動状況>

- 着想の段階 (アイデア・構想の発表)
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他 ()

<発表内容>

1.背景

最近、アジャイルはソフトウェア開発だけでなく、ハードウェアや製品、さらにマネジメントや組織論の場でも語られることが増えてきた。では、アジャイルとは何なのか？私たちは、アジャイルは価値を最大化するための「考え方」であり「姿勢」であると考える。つまり、製品の価値だけでなく、ビジネスの価値・組織の価値も最大化することができるはずである。私たちは舞台を日新システムズに置き、2015年10月、未来戦略室という部門を立ち上げ、アジャイルな組織改革に取り組んでおり、その活動内容を紹介する。

2.改善したいこと

弊社、日新システムズは受託開発中心で成長してきたが、さらに大きな成長のためのシフトチェンジを目指している。それは、「これ作ってね！」という受け身な受託開発から、顧客価値提案型の開発へと変化し、営業と技術が協調しながら、新規ビジネスモデルを構築できる未来への価値を創りだせる強い会社へと成長することである。

未来戦略室は、アジャイルの考え方をベースに、弊社の目指す未来を明確にしつつ、個々の知識を増やし、つなぎ合わせ、「意欲ある自律した創造的な社員」が集まる「場」に変えていく！ それが私たちの活動のコアにあるものである。

3.改善策を導き出した経緯

今までアジャイルを導入してきた経験から、成功するためのチェックポイントを挙げてみた。

- ✓ プロジェクトにエンジニアリング＆プロセスの基本スキルはどのくらい？
- ✓ メンバが風土を変えるぐらいの改善意欲を持っていますか？
- ✓ 自分たちが作っているものに愛着を持っていますか？
 - お客様が何を求めているのか考えてる？
- ✓ 改善指標を数値だけで判断していませんか？
 - コミュニケーションは活発？
- ✓ メンバとゴールを共有できていますか？

そこで、弊社の現状を分析してみたら、こんな結果になった。

<チェックポイント>	<現状>
プロジェクトにエンジニアリング＆プロセスの基本スキルはどのくらい？	一人ひとりのスキルは高く、土壇場でスーパーマン的な頑張りでピンチを乗り越える。 でもプロジェクトを運用するツボ（スキル／マインド）が不足がち。
メンバが風土を変えるぐらいの改善意欲を持っていますか？	品質保証体制はしっかりと構築されており、プロセス監査活動が実施しているが、形骸化が進み、自分たちでプロセスを設計し、自分たちで改善する方法がわからない。失敗しても次に応用できるんだ！という風土になってない
お客様が何を求めているのか考えてる？	お客様から要求されたことはできるが、 お客様の抱えている課題を先につかんで提案できる人が少ない
コミュニケーションが活発？	一人作業が多く、知識・ノウハウが共有できていない。 形式的な組織はあるが、「チーム」、「チームワーク」が少ない！
メンバとゴールを共有できていますか？	トップダウン型のリーダーシップが強く、現場はやらされ感が蔓延しがち。失敗しても次に応用できるんだ！という風土になってない

これらの課題を解決する上で、未来戦略室としては、SECI モデルやアジャイルのスクラムをあてはめながら、自己組織化を目指す戦略を考えてきた。

目標は、会社の目指す未来を明確にしつつ、個々の知識を増やし、つなぎ合わせ
日新システムズを「意欲ある自律した創造的な社員」が集まる場に変えていくことである。

4.改善策の内容

改善策を、未来戦略室の3つの柱としてまとめる。

① 値値創造

- 経営成果の定量的分析
- ビジネス戦略と組織開発の連携

② 組織改善

- SEPGとのプロセス改善・推進
- 風土改革活動推進

③ 人財開発

- スキルマップ、キャリアパスの再定義
- 戦略的トレーニングの実施と推進
- 「場」の設立：社内ワーキングルーム設立

5.改善策の実現方法（代表的な活動内容の紹介）

① 値値創造

- 会社のゴールを描いて共有する
 - ・ 中期計画策定および部門方針策定：経営幹部、管理職をファシリテートして、ゴールへのコミットを形成。
策定後は半期毎にPDCAサイクルを回し、一緒に振り返り、経験から学ぶ機会として運用する。わくわく
社員がワクワクするような活動目標をつくり、部門全体で共有する働きかけを支援

② 効率のよい開発への改善：自分たちのこととして改善していく+SEPGによる改善推進

- 各事業部にSEPGという役割を新たに作り、活動支援
- Agile導入プロジェクトの支援
- 全プロジェクトを対象に「ふりかえり/KPT」実施を運営
- CMMIを共通モデルとして導入 ⇒内部アブレイザルを実施し改善につなげる

③ 人財開発

- 管理職塾の開催：管理職の意識改革
 - ・ 弊社は「トップダウン型マネジメント」が強い傾向がある。メリットもあるが、現時点では「やらされ感」「思考停止」などデメリットな面が強く出ている。そこで、管理職が職場風土の改善を考える「場」として管理職塾を立ち上げ、運用を始めた。現場の課題を見つけ、真因を分析し、解決することがゴールだが、合わせて、自分たちに何が不足しているか、気づきを持ってもらうことも目的である。ファシリテーションやコーチング、傾聴といったスキルを改めて学び、現場で実践することで習得してもらうようなプログラムで推進中。
 - ・ リーダーシップ塾を次世代リーダーの育成のために開催する予定
- コワーキングルーム設立！
 - ・ 強い「意欲」で「未来」を創りだせる人の、学びたいという「意欲」をカタチにする「場」
 - ・ 社内外の交流の場として活用。朝会、勉強会、方針発表会、休憩、懇親会などなど、コミュニケーションの活性化に貢献

6.改善による変化や効果

<チェックポイント>	<変化／成長の兆しが出てきた>
プロジェクトにエンジニアリング＆プロセスの基本スキルはどのくらい？	SEPG が各プロジェクトを支援し、徐々に現場の本音が噴出！本音から課題を抽出し、改善案を組織の上位層に提案し、改善活動を展開中！アジャイル導入プロジェクトが増加中。次世代リーダーが中心になって、少しずつ成功体験を積み重ねている。
メンバが風土を変えるぐらいの改善意欲を持っていますか？	「ふりかえり」では、良かった点(K)を見つけること、弱み(P)については改善案(T)を全員で考えることに重点を置いたファシリテートを実施。「失敗＝責められる」の意識が変わってきた。
お客様が何を求めているのか考へてる？	各事業部に事業企画室を立ち上げ、「お客様から要求されたものを作る開発」からのシフトチェンジを推進
コミュニケーションがは活発？	朝会やふりかえりを通じて、コミュニケーションの大事さに気づき、チーム作りへのアプローチが始まる。朝会も一部で始ましたが、全社に展開中。
メンバとゴールを共有できていますか？	管理職が、管理職塾でのディスカッションを通じて、信頼関係の重要さを意識し始め、特に「思いの伝え方」を課題として抽出→要望に応えコーチングセミナーを開催！

<残課題と今後の取組み>

- 活動の成果として、変化は一部のメンバーに現われてきた。
 - ・ 「改善」が、少しずつ開発現場に浸透してきた
 - ・ 次世代リーダーはアジャイルを導入して、成功体験を積んでいる
 - ・ コミュニケーションは、以前より活発になってきている
 - ・ 管理職（部門長）の意識も変わりつつある
- でも、いくつかのプロジェクトで内部アフレイザルを実施してみると、改めて見えてきた課題があった
 - ・ 管理職の意識は変わりつつあるが、「やり方」は変わらない
 - ・ 部門長以上のマネジメントスキルが不足している
 - ① 中身のあるプロジェクト計画が作れない
 - ② 場当たり的なマネジメント中心
 - ③ 発生した課題に的確に対応できない
 - ④ プロジェクト完了後、ふりかえりができないのでスキルアップしない
 - ⑤ マネジメントスキルが不足したまま →①へ 負のサイクルが続く
- 全社のマネジメント強化対策→努力と根性だけでなく「考える」事業部への変革を促進
 - ・ 未来戦略室と SEPG がプロジェクトに入り込み、座学ではなく、実践の場でスキルアップを図る
- 今は基盤作りのステージ
 - ・ 短期間でしっかりと基礎を築くために、トップダウン＆ボトムアップのアプローチを同時に実践。

<まとめ>

『ワクワクする未来を創造する』 アジャイルな組織を目指すツボとは？

未来戦略室にとって、お客様は日新システムズであり、その価値を最大化することがミッションである。そのため、私たち自身がまずアシリティに取り組むことが重要だと考えています。

- 組織／プロジェクトにリズムを作る

- ゴール（経営目標）を共有し、現場との相互のフィードバックを行う
 - “今”の組織／プロジェクトを分析し、“今”の最大の価値を引き出す戦略を考え、実践を支援する
- これらの活動をベースに、今後も日新システムズが「自分たちで発展できる組織」へ変わるために戦略を進めていきます。

7.改善活動の妥当性確認

一連の取組みにより、一部ではあるが、社員の意識が変わってきていることは確かであり、それぞれ行動の変化が表れてきている。これからはその変化をチームや組織に向け、結集していくことが課題である。

2015年10月から始めたばかりの活動であり、今後、本活動の成果を評価していきたい。

A.参考文献

<タイトル>

プロセス改善活動をスムーズに立ち上げるための取り組み<その2>

<サブタイトル>

～改善活動初心者のあなたが CMMI 公式評定メンバとして参加することになったら～

<発表者>

氏名 (ふりがな) : 中村英恵 (なかむら はなえ)

所属 : 株式会社 NTT データ

<共同執筆者>

氏名 (ふりがな) : 大川瑠里子 (おおかわ るりこ) , 木暮雅樹 (きぐれ まさき) , 矢部 智 (やべ さとし)

所属 : 株式会社 NTT データ

<要旨・主張したい点>

NTT データでは組織のプロセス改善活動の立上げをスムーズに質よく実施することを目的として「改善活動 Startup」を作成し展開している。

しかし実際には、改善活動には評定の準備段階からメンバが新たに参加する場合や、CMMI やプロセス改善活動の初心者が参加することもありうる。

限られた期間において CMMI ベースの改善活動を円滑に進めるためには、初心者であるメンバが参加の初期段階で活動の流れを把握し、タスク実行の際には CMMI のモデルの理解をフォローする取組みが不可欠である。

本発表では、CMMI もプロセス改善活動もどちらも初心者のメンバが CMMI 公式評定メンバとして参加した事例を背景に、「改善活動 Startup」の改善を通じて評定フェーズにおける活動の流れやタスク遂行におけるモデル理解のための情報整備に向けた取り組みとその効果について報告する。

<キーワード>

プロセス改善活動、初心者、立ち上げ、ノウハウ有形化、ベストプラクティス、CMMI

<想定する聴衆>

組織の SEPG、CMMI に基づく改善活動の初心者

<活動時期>

2016.4～

<活動状況>

- 着想の段階 (アイデア・構想の発表)
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他 (変更を実施中 (未完) であり結果も明確ではない)

<発表内容>

1. 背景

筆者の所属するチームでは CMMI に基づく改善活動の支援を実施している。本活動では次の 3 つのフェーズを定義する。

フェーズ 1 :「立上げ・ギャップ分析」

フェーズ 2 :「アクションプラン実施」

フェーズ 3 :「評定」

これまで、支援の一環として、ギャップ分析フェーズにおける新規参入メンバ向けのタスク実施手順や様式を備えた「ギャップ分析フェーズ Startup」を作成し、展開している。

筆者はコーポレートの SEPG という立ち位置で、評定フェーズ活動へ入ったばかりの社内支援先組織において、公式評定メンバとして参加することとなった。筆者自身は CMMI ベースのプロセス改善活動は初心者である。また、評定フェーズ活動の Startup を作成する役割を担う。

支援先組織の SEPG には、公式評定メンバとして新たに 2 名加わっている。いずれも筆者同様に CMMI ベースのプロセス改善活動は初心者のメンバである。

筆者が参加時、支援先組織の評定フェーズ活動では、組織 SEPG メンバによる PIIDs 収集活動の段階であった。筆者はレディネスレビューで報告するプロセスエリアの担当を割り当てられ、組織 SEPG メンバによる PIIDs 収集結果を確認することになった。ギャップ分析フェーズとは異なり、評定フェーズでは PIIDs の根拠の説明が求められる。

CMMI に基づく改善活動の経験が乏しいメンバの存在は、評定フェーズを期間内に成立させることのリスクとなりかねない。

活動の進捗の妨げになることなく、効率よく各タスクを実施するためには、立ち位置に限らず初心者メンバの CMMI に基づく改善活動および評定フェーズの活動の理解と CMMI のモデル理解をいかにスムーズに促進するかが不可欠である。

また、組織の自律的改善という観点では CMMI ベースのプロセス改善活動の全体像を明らかにした上で、新規に参加したメンバの現在地が把握できるよう理解を促すガイドが必要である。今回の背景のように評定フェーズから参加するメンバもいることを勘案しても評定フェーズ Startup のニーズは高いと考えられる。

本発表ではギャップ分析フェーズ Startup の改善を通じて、初心者が CMMI に基づく改善活動を遂行するために必要となる「活動そのものの理解」とタスク遂行上の「モデルへの理解」への足がかりとなる手順書に求められる要素を整理し報告する

2. 改善したいこと

(1) ギャップ分析フェーズ Startup の改善

ギャップ分析フェーズ Startup は各組織に展開しているが、現行 Startup では想定外の CMMI-SVC のモデル概説を求められたり、国内に複数拠点を持つ組織においては組織の事情を考慮したギャップ分析手順を再編集する必要があるなど、改善活動 Startup をとりまく状況は変化している。

(2) タスクを遂行する上で必須となるモデルへの理解

社内組織のレディネスレビュー準備のために組織内 SEPG が収集した PIIDs 一覧を渡された。

モデルに関する解説資料はあるが、今回収集された具体的な成果物がどうしてプラクティスと対応するのか、根拠であろうと思われた認識が正しいのかどうか確認するにはチーム内経験者への確認が必要であった。

レディネスレビューにおいては、新規参加メンバは LA からの指摘に対して PIIDs の根拠の説明に苦慮する場面も見られた。

3. 改善策を導き出した経緯

(1) ギャップ分析フェーズを多様な組織に適用する上で発生した改善課題は次の 4 つであった

- ・タスクの流れがすべての組織に当てはまるとは限らない
- ・CMMI モデルの解説とギャップ分析手順の関係がとりづらい
- ・どのコンテンツを重点的に読んだらいいのかわかりづらい
- ・ギャップ分析結果を踏まえたアクション立案について考え方の指針がない

これらの課題に対し原因と対策を検討した結果、次の 4 つの対策を講じることとした。

対策① コンテンツを分類する

対策② 他組織事例を踏まえてタスクの粒度を見直す

対策③ 説明や事例を追加する

対策④ 手順のパート化

なお、コンテンツの分類に関しては以下を考慮することとする。

タスクについては、PMBOK「スコープマネジメント」の要素の一つである WBS の作成に着目して確認する。

・タスクの分類

タスク一覧表には、タスクが概ね大分類、中分類、小分類に分かれている。

現在、大分類は 1 つあたりに対応づけられる中分類項目の平均値は 3.7 であり、最大値、最小値、中央値では最大値は 8 つ、最小値は 1 つ、中央値は 3 つであった。

このことから、大分類 1 つあたりの中分類の数にはばらつきが見られ、多数の中分類が対応付けられている。大分類はさらにタスクグループを分割できる可能性がある。また、1 つまたは 2 つしか中分類のない大分類は統合について検討の余地がある。

・タスクの粒度

小タスクでは、ワークパッケージとしての最小単位としての実施事項が書かれている。

具体的には、「作成する」「(PIIDs を)収集する」「レビューする」「データを共有する」「説明する」「対象範囲を決定する」などである。1 つのタスク完了に要する時間の観点で比較すると、「PIIDs を収集する」という一定の期間が必要なタスクと「データを共有する」や「対象範囲を決定する」という比較的すぐに終了するタスクが同じレベルで記載されており、一定のタスク粒度の目線でみることが難しい。

・他のドキュメントとの整合性

活動全体の流れを解説している他の説明資料とタスク一覧とでは大分類のタスクレベルで名称やタスクの対応づけで不整合が見られた。活動全体の流れを理解しタスク一覧で詳細を確認するに際し不一致部分が発生したために表の中の位置そのものを見失い、現在自分が実施しているタスクがどれに該当するのかわからなくなってしまった可能性が高い。

(2) タスクを遂行する上で必須となるモデルの理解

レディネスレビューでは、PIIDs として収集した証跡を根拠とともに評定チームメンバに対して説明する必要がある。

組織 SEPG の新規参加メンバは PIIDs 収集時には、既存の PIIDs の最新化が作業の中心のことであった。

これでは PIIDs として対応付けられている各成果物やドキュメントを日付の新しいファイルに置き換えているだけであり、プラクティスに対する成果物に求められる要件が何であるかの理解を補うことができない。仮に、新規メンバーは PIIDs を最初から集めなおす、というアプローチであったとしたら、所要時間も相当必要であることが見込まれ、限られた期間で評定フェーズの活動を終えることができない。何よりも、本来目的である組織の自律的な改善活動を遂行することができない。

評定フェーズ Startupにおいては、上記改善点の対策を踏まえて作成するとともに、ギャップ分析フェーズ Startup についても改善する。

4.改善策の内容

(1) ギャップ分析フェーズ Startup は次の方針で組織に適用し、編集した場合は事例として蓄積する

1. 組織の事情に合わせて必要に応じて情報を抽出し、再編集して提供する（④手順のパート化）
 - 改善活動 Startup（立上げ・ギャップ分析フェーズ）は豊富な情報源
2. CMMI も改善活動も初心者のメンバーに対しては、ギャップ分析手順説明のまえに、実際のプロジェクトの成果物（プロジェクト計画書）を引き合いに、CMMI との関係を説明する。（③説明や事例を加える）
3. タスクで前提となっている考え方（インタビューの対象や実施単位など）については初めて改善活動を実施する組織に対しては予め説明する。（③説明や事例を加える、④手順のパート化）
4. ギャップ分析後のアクションプラン検討に向けて、弱みや改善の機会に対するアクション方針案の整理について事例を蓄積する（④説明や事例を加える）

(2) 初心者向けに評定フェーズのタスク一覧を作成する

評定フェーズの流れは SCAMPI トレーニングで学ぶことができるが、本活動では評定フェーズ活動を実施するために必要な具体的な事項を事例や様式とともに提供することで、初心者への知識の補完をめざす。

一般に、プロジェクト管理における WBS 作成においては次のルールやコツがあるといわれている。

- 「子要素は親要素を 100% 満たすものでなければならない」
- 「1 つのワークパッケージが概ね 8 時間以上、80 時間以内になるまで分解する」
- 「1 つの親要素にぶら下がる子要素の数は 7 つ程度まで」

そこで、評定フェーズ活動の Startup としてタスクを整備する際には、これらのルールを考慮する。

また、最大 7 つを目安にフェーズ活動を大分類し、評定フェーズ活動は週間進捗を行うものとして仮置きして小分類のタスク粒度を 1 日から 10 営業日程度でおさまるものを目安として抽出し構造化する。

ただし、PIIDs 収集など一定の期間を要するものについては例外としこの限りではない。

なお、大分類への展開時には、活動の中のまとまりとして区切りがつけやすいものを考慮するほか、タスク名称や分類が、他のでは、同一の語句を含まないよう配慮する

さらに、ギャップ分析フェーズの Startup についても同様の見直しを行い、タスク粒度を合わせる。また、関連ドキュメントとの整合性をチェックする。

(3) タスクを遂行する上で必須となるモデルの理解を補うには

モデルの理解が最も必要とされるのは PIIDs 収集である。後のレディネスレビューにおいて根拠を説明する必要があることから、収集担当者の理解の補助となる、各プロセスエリアのプラクティスに成果物を対応付ける際の確認ポイント表を作成し提供する。

5.改善策の実現方法

(1) ギャップ分析フェーズの Startup の改善に向けて

2つの組織に対して、改善策を考慮してギャップ分析フェーズ Startup を適用する。

(2) 初心者向けに評定フェーズ Startup 作成に向けて

評定フェーズ活動のタスク一覧では、WBS 作成ルールに従って具体的なタスクを抽出し分類する。ギャップ分析 フェーズでのタスク一覧についても同様の抽出を行い、既存のタスクと比較しタスク粒度を調整する。

まず、成果物ベースでみたとき、評定フェーズ活動の第 1 の要素は「評定計画書の作成」と「評定結果」の 2 つに分かれ。しかしこれでは単位が大きすぎるため、作業ベースの分解を試みる。

ギャップ分析フェーズの Startup 資料の一つである「改善活動全体の流れ」を記したドキュメントでは活動を次の 6 つに分けている。

「評定計画情報の収集」

「計画策定/体制確立」

「評定準備/トレーニング」

「証跡収集/その他準備」

「レディネスレビュー」

「評定実施」

先に示した WBS 作成ルールの 1 つである“要素レベルは 7 つまで”に合致している。また、まとまりのある区切りとしても経験者により区分されたものであるため、これを大分類としてタスクを詳細化する。

小タスクの抽出は作業ベースで行うこととし、「格納する」「承認する」「説明する」といった 1 日以下を目安に実行可能と思われるタスクは、小タスクの完了条件として位置づける。

たとえば「インタビュー項目を作成する」というタスクなど、そのタスク自身の進め方を記した手順が必要となる場合がある。その場合はその小タスクに紐づく手順として位置づけ、整理する。同様に、既存の様式、流用可能な様式は小タスクと関連付ける。

ギャップ分析フェーズでは存在しないタスク（レディネスレビュー以降）では、経験者有識者のインタビューを実施したうえでタスク分解する。その際も WBS 作成ルールを考慮した分解を行う。

(2) タスクを遂行する上で必須となるモデルの理解を補足する

現在、ギャップ分析フェーズの Startup では PIIDs 収集の際の各プラクティスのポイントを示した資料がない。

レディネスレビューでは、サブプラクティスに書かれた事項との対応について説明が求められる場面が多く見られた。

そこで、取り掛かりとして、プラクティスごとのサブプラクティスまでの一覧表を作成し、既存のモデル解説資料やレディネスレビューにおける指摘内容を踏まえて PIIDs の根拠として何が必要だったかをプラクティス別に補完する。

経験者、有識者を交えて PIIDs 収集のポイントとしての過不足についてレビューを実施する。

レディネスレビュー実施の都度、確認ポイントをブラッシュアップさせていく。

6.改善による変化や効果

(1) ギャップ分析フェーズ Startup の改善

1. 示した手順どおりにギャップ分析を完了できた
2. プロセス領域説明資料はデータ整理統合時に活用できた
3. ギャップ分析の次の二フェーズにつなげる情報提供のポイントを獲得

(2) 評定フェーズにおける活動やタスクの理解度について(今後実施)

評定フェーズ Startup 配布時に、更新版情報としてギャップ分析フェーズ Startup も配布する。

評定フェーズ Startup の配布先には改善活動の流れや評定フェーズ活動の流れと進めやすさについてアンケートを実施する。

配布時に受けた質問のうちタスクの内容や段取りに関するものがどの程度発生したかを集計し分析する。

(2) モデルの理解度向上について（今後実施）

PIIDs 収集前に評定フェーズ Startup の配布し、レディネスレビューでの所要時間、指摘事項、追加で必要となった PIIDs の数をカウントする。

評定フェーズ Startup を配布せず、新規メンバが参加したレディネスレビュー時の実施結果と上記を比較し検証する。

7.改善活動の妥当性確認

今後、別組織での CMMI をベースとした改善活動支援を実施する際に、評定フェーズ Startup を配布し本改善策の妥当性を確認する。更新版情報としてギャップ分析フェーズ Startup も配布する。

直近の支援予定先は複数の部門からなる統括された組織である。

統括グループの SEPG は部門の SEPG を束ね、コーポレート SEPG の窓口となる役割をもつ。

妥当性確認は次の手順で実施する。

1) 今回作成する評定フェーズ Startup(プラクティスの確認ポイント表を含む) を

- (a)コーポレート SEPG から統括グループ SEPG に展開する。
- (b)統括グループ SEPG は部門 SEPG に評定フェーズ Startup を展開する。

2) 次の情報を収集する。

- ・(a)で発生した質問の数
- ・(b)で発生した質問の数
- ・SEPG 別のレディネスレビューでの指摘事項の数

A.参考文献

PMBOK

「WBS には、守るべき 3 つのルールがある 」

安達裕哉、IT メディアエンタープライズ連載 「WBS でプロジェクトを成功させる（2）」

<http://www.itmedia.co.jp/im/articles/1001/27/news103.html>

<タイトル>

プロセス分析に基づくドキュメント再構成によるプロセス改善

<サブタイトル>

車載通信モジュール開発における実施と効果予測

<発表者>

氏名 (ふりがな) : 小島 裕次 (こじま ゆうじ)

所属 : 株式会社デンソー 情報通信技術 3 部 第 4 技術室

<共同執筆者>

氏名 (ふりがな) : 古畠 慶次 (こばた けいじ)

所属 : 株式会社デンソー技研センター 技術研修部 企画室

<要旨・主張したい点>

これまで社内では、CMMI や Automotive-SPICE などのモデルを使用したプロセス改善を推進してきた。しかし、モデルによる改善は推進者が現場のプロセスよりも、モデルへの準拠を重視したため、現場に定着するのが難しかった。

そこで、我々は現場のプロセス分析に基づいたプロセス改善を実施した。具体的な方法は次のとおりである。最初に、不具合情報の分析とプロセス分析から、問題のドキュメントを特定する。そして、問題のドキュメントに記載されている情報の有用性を分析し、分析結果に基づいてドキュメントを再構成することによりプロセスを改善する。以上により、現場が受け入れやすいプロセスを設計でき、ドキュメントの品質改善、コスト削減の効果が期待できることを確認した。

<キーワード>

プロセス分析、ドキュメント再構成、プロセス設計、コスト削減

<想定する聴衆>

マネージャ、SEPG、SQA、現場の設計技術者

<活動時期>

2015 年 6 月～現在

<活動状況>

- 着想の段階 (アイデア・構想の発表)
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他 ()

<発表内容>

1.背景

我々が開発を担当する車載通信モジュールは 2013 年モデルから車内ネットワークに接続され、リモート診断機能やリモート制御機能が追加された。現在も機能追加が続いている。年々ソフト開発規模および開発に関わる技術者の数が増加している。また、開発期間は、市場ニーズの高まりにより徐々に短くなっている。

これまで、CMMI や Automotive-SPICE などのモデルを使用したプロセス改善を推進してきたが、根本的な問題の解決ができない、開発コストの増大、品質の低下が大きな問題になっている。

2.改善したいこと

現状のままでは開発規模の増大には対応できず、工数不足により品質の低下に歯止めがかかる可能性が高い。しかし、現場の技術者は、問題は上流の要件定義工程のドキュメントにあることは認識していたが、何が問題かについては十分把握できていなかった。

そこで、不具合情報から問題のプロセスを明らかにし、対象のプロセスを分析する。これにより問題の原因を明らかにし、成果物であるドキュメントに着目することでプロセス改善を実施する。

3.改善策を導き出した経緯

プロジェクトで発生している不具合を分析することで、要件定義工程に問題があることを特定した。次に要件定義工程に対して PFD を用いてプロセス分析し、ソフトウェアシステム設計書(SS 設計書)が次工程で有効に使われておらず、SS 設計書と同じ入力文書に基づいて外部仕様書が作成されていることが分かった。

SS 設計書と外部仕様書の内容について分析したところ、要求から仕様が導出されておらず、この結果、要件定義に起因する不具合が発生していることが分かった。

この状況を改善するために、SS 設計書と外部仕様書を再構成することで、要求仕様が導出されるプロセスへ改善することを着想した。

4.改善策の内容

改善策を決定する方針として、モデルを使用した改善の反省に基づき、現場重視の改善を進めるため、次の 3 点を掲げた。

- (1) 現場の負担をなるべく軽くする
- (2) 無駄に成果物を増やさない
- (3) 既存の成果物を徹底活用する

改善策は、既存の成果物を極力利用する形で再構成する方法を検討した。既存成果物のどの部分が後工程で活用されているか、現場の意見を集め、活用している情報はそのまま採用し、活用していない情報は削除することで、成果物の情報を整理する。

整理した成果物の情報を、組込みソフトウェア向けのプロセス標準と対応をとることで、現状の成果物を再構成する。プロセス標準に対して不足している情報は、極力現場の負担にならないよう追加を検討する。

5.改善策の実現方法

(1) 成果物の活用度調査

問題の成果物が後工程でどれだけ活用されているかを、貢献度という指標で表現した。貢献度は現場の担当者からヒアリングし、その活用度合いに応じてポイント化して決定した。

(2) 成果物の情報整理

成果物間で重複している項目について、貢献度の高い内容はそのまま採用した。貢献度の低い内容はどんな

タイミングでどの部分が活用されているか調査し、その結果に応じてマージするか、削除するかを個別に検討した。これにより、問題の成果物の情報を整理した。

(3) ドキュメント再構成

ドキュメント再構成にあたっては、組込みソフトウェア向けのプロセス標準として、組込みソフトウェア向け開発プロセスガイド(Embedded System development Process Reference:ESPR)を採用した。ESPRを車載通信モジュール向けにテーラーリングし、整理した成果物の情報を対応させることにより、成果物を再構築した。

6. 改善による変化や効果

問題の成果物であるSS設計書と外部仕様書を、ソフト要求仕様書とアーキテクチャ設計書に再構築した。これにより、要件定義プロセスの入出力のドキュメントが整理でき、正しく要求仕様が導出できるプロセスに改善することができた。

改善前後のドキュメント品質は、斎藤らが提案している RISDM (Requirements Inspection System Design Methodology)を適用して評価を実施した。その結果、主に責任追跡性、明確性、記述網羅性の視点で大きく品質が改善されることが確認できた。

コスト削減効果については、成果物の品質が向上したことで、要件定義に起因する不具合の低下が見込まれる。RISDM の評価結果から、要件定義に起因する不具合が約 55%削減されたと仮定した場合、不具合に対応する手戻り工数が約 26%削減できることを確認できた。また、RISDM の評価指標である明確性と記述網羅性の品質が改善されたことで、コスト予実乖離リスクの改善が見込まれる。

7. 改善活動の妥当性確認

今回のプロセス改善によって、要件定義工程が機能するプロセス設計ができたことにより、要件定義に起因する不具合を削減する目途がついた。また、既存のドキュメントを活用して成果物の再構成を実施したため、現場への負担増加は最小限に抑えることができる。本改善に必要な現場の工数は、不具合削減による手戻り工数の低減を考慮すれば十分見合うものである。

A. 参考文献

- [1] 独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター(編), 改訂版 組込みソフトウェア向け開発プロセスガイド, 初版, 翔泳社, 2007
- [2] 斎藤忍 他, RISDM:ソフトウェア要求仕様書のインスペクションデザイン方法論の提案と適用評価, SES2014, Sep. 2014, pp. 105–114.
- [3] 清水吉男, [入門+実践]要求を仕様化する技術・表現する技術～仕様が書けていますか?, 初版, 技術評論社, 2010.
- [4] 清水吉男, 「派生開発」を成功させるプロセス改善の技術と極意, 技術評論社, 初版, 技術評論社, 2007.

1B3 「プロダクトライン開発におけるプロセスのコア資産フィードバックモデルの提案」 林健吾(デンソー)

<タイトル>

プロダクトライン開発におけるプロセスのコア資産フィードバックモデルの提案

<サブタイトル>

車載ソフトウェア派生製品開発プロセスの立ち上げ事例

<発表者>

氏名 (ふりがな) : 林健吾 (はやしけんご)

所属 : (株)デンソー

<共同執筆者>

氏名 (ふりがな) :

所属 :

<要旨・主張したい点>

プロダクトライン開発において、アプリケーション導出が繰り返されることを利用して、プロセスをコア資産に登録してフィードバックするモデルを提案する。各導出で生成される成果物に HOW・WHY を残すことで、プロセスを再利用し易くなるよう形式知化を促進する。開発を監督するアーキテクトが、組織の暗黙知を収集し、標準プロセスと段階的に無理なく統合していくことで、形式知化がさらに促されるよう工夫した。結果として、新たに立ち上げた組織のアプリケーション導出プロセスを確立し、製品リリースすることに成功した。

<キーワード>

プロダクトライン開発、SPLE、プロセスライン、プロセス改善、派生開発、暗黙知、形式知、フィードバック

<想定する聴衆>

ソフトウェアエンジニア、プロダクトライン開発者、組込系技術者、SEPG

<活動時期>

2015年1月～2016年4月（以降、継続中）

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

我々は、走行支援センシングシステムのソフトウェアを開発している。このシステムは、技術競争が活発で市場は拡大傾向にある。そのため、第1世代のシステムを先頭車両に投入後、すぐに第2世代へと機能を向上させながら、第1世代多くの車両に拡販・展開する。本開発に対応すべく、我々の組織ではプロダクトライン開発に取り組み、世代を進化させるコア製品開発組織と、コア資産から派生製品を数多く導出する車両展開開発組織の2つの組織体制を採用した。主体は車両展開開発組織であり、発表者は主体のリーダーを担う。組織からの主体の立ち上げ要請を受け、活動を開始した。

2.改善したこと

最初の派生製品リリースまでは1年の期間があり、その後は年間4~6製品リリースしていく。この間に、製品の品質を確保して製品リリースできる組織を立ち上げたい。しかし、チームメンバー7名の内、開発領域への知見を有しているメンバーは発表者を含めて2名だけであり、知見は十分でない。組織の標準プロセスも派生製品の導出には最適化されていなかった。この状況において、最初の派生製品リリースまでに、組織監査を通過できるレベルの派生製品導出プロセスを確立し、その後に続く多数の派生製品リリースへの体制を構築することが課題となっていた。

3.改善策を導き出した経緯

プロセスと知見(知識)をキーワードに、関連研究を紐解いて改善策を導いた。

- ・プロダクトライン…プロセスをコア資産に登録し、アプリケーション導出時に適用するプロセスラインが提案されている
- ・プロセス改善の方法…標準からプロセス定義するトップダウン方式と、現場から共通性を見出すボトムアップ方式がある
- ・知見不足…暗黙知を形式知化することで組織知識を成長させる SECI モデルが提唱されている

最初の派生製品リリースまでには、その他の車両向けを含め、試作製品リリースが6回あることがわかつっていた。この間に、改善のフィードバックが掛かるようにプロセスを資産化して再利用すれば、回数を重ねることでプロセスの改善レベルが一定レベルまでは向上して、派生製品導出プロセスが確立できると仮定して改善策を立案した。

4.改善策の内容

プロセスは固定するのではなく、派生製品の導出を繰り返す中で成長させるものとして定義した。プロセスを成長させるフィードバックモデルを以下に示す。

0. 組織標準プロセスを参照しながら、派生製品導出用のプロセスを抽象度高く設計し、初期プロセスセットを構築する
1. 構築プロセスをプロセスラインとしてコア資産に登録する
2. 派生製品導出において開発フェーズ（試作レベル・最終版）に応じてプロセスをコア資産から引き出す
3. 派生製品導出時にプロセスを具体化させながら実行する
4. プロセス実行時に生じた問題を振り返ってプロセス定義を改善する
5. 改善したプロセスをコア資産として登録する
6. 2~5を繰り返す

5.改善策の実現方法

改善策を実現するために、以下に示す3点の工夫を取り入れた。

1. プロセスのコア資産の監督者の任命

コア資産におけるプロセスの監督者としてアーキテクト（今回は発表者自身）に責務を与えた。アーキテクトは、コア製品開発組織の有する暗黙知を集め易く、形式知としての組織標準プロセスにアクセスし易いメンバーを割り当てた。プロセス実行前の計画時と、成果物を生成した後のレビュー時、コア資産への登録時において、コア資産としてのプロセスの妥当性を確認し、プロセス定義の修正方法を決定する。

2. フィードバックと再利用しやすいプロセス定義の規定

資産としてプロセスが活用され易くするため、手順を記述するプロセス定義やテンプレートを充実するのではなく、成果物を再利用参照対象として記録する方針とした。成果物には単なる結果だけではなく、方針や理由、判断方法、設計方法など、HOWとWHYを表現し、再利用時に手本にできることを、アーキテクトの確認ガイドラインとした。

3. 段階的なプロセス改善

はじめからプロセスの完成を目指すのではなく、最終製品リリースまでに段階的に改善していくよう工夫した。試作製品開発では組織標準プロセスからテーラリングで大きく省略して始め、改善の必要に応じて足し合わせるようにした。具体的には、初回成果物生成から、ダブルチェック・クロスチェックの導入、チェックリストの導入、レビュー・チェックシートの導入、一部作業の自動化、といったように、品質確保を優先してから効率化を図る方針で改善を目指した。

6. 改善による変化や効果

改善策を実施した結果、派生製品導出を繰り返す度に、再利用されるプロセスで登録される成果物が成長していくことが観察された。また、それぞれのプロセスに専門の開発者が担当するのではなく、開発ごとに開発者をローテーションしても、作業時間の上下があるだけで、品質の安定した成果物を得られることが認められた。成果物を参照したプロセスが再利用される頻度を計測したところ、85%のプロセスが1回以上再利用される結果が得られた(図1、2参照)。また、2週間単位の開発量を計測したところ、開発が繰り返されるに従って、開発量が向上しながら安定に向かう結果が得られた(図3参照)。プロセスが確立されて、安定的に実行できるようになっていったと言える。

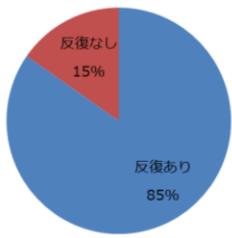


図 1.

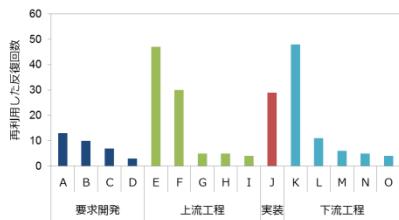


図 2.

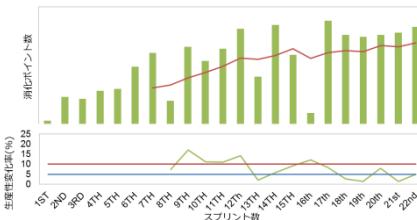


図 3.

7. 改善活動の妥当性確認

プロダクトライン開発やその他派生開発のように、特定のプロセスが繰り返され、再利用の機会の多いプロジェクトにおいて本改善策が有効に働くと考察される。成果物を直接手本として参照でき、設計や調査方針や判断方法が残されたことも、再利用が促進された要因と考えられる。但し、成果物にHOWやWHYを埋め込むことから、品質が安定する代わりに、生成の速度は落ちる。各プロセスにかかる時間を計測し、重たいプロセスを優先的に自動化するなどのさらなる工夫が必要である。

A. 参考文献

- [Romb05] D. Rombach, "Intergrated Software Process and Product Lines", Post-Proceedings of Software Process Workshop 2005, LNCS Vol.3840, 2005
- [Haya06] 林好一, ソフトウェアプロダクトラインエンジニアリングをプロセスステーラリングに応用する, 第25回ソフトウェア品質シンポジウム, 日本科学技術連盟, 2006
- [Fuji13] 藤繩幾子, レベル3達成組織における自律改善の推進, SPI Japan 2013, JASPIIC, 2013
- [Nona96] 野中郁次郎, 知識創造企業, 東洋経済新報社, 1996

<タイトル>

ビデオ撮影及び画面キャプチャによる分析プロセスの改善

<サブタイトル>

<発表者>

氏名 (ふりがな) : 原田 大輔 (はらだ だいすけ)

所属 : 住友電工情報システム株式会社 QCD 改善推進部 品質改善推進グループ

<共同執筆者>

氏名 (ふりがな) : 中村 伸裕 (なかむら のぶひろ)

所属 : 住友電工情報システム株式会社 QCD 改善推進部

<共同執筆者 2>

氏名 (ふりがな) : 服部 悅子 (はっとり えつこ)

所属 : 住友電工情報システム株式会社 QCD 改善推進部 品質改善推進グループ

<要旨・主張したい点>

弊社では、Java によるプログラム開発の品質向上のために自動テストを導入した。その結果、品質は良くなつたが、一部プログラマの生産性が下がってしまった。

プログラマに作業時間を 5 分単位で記録してもらったところ、テスト設計に問題がある事が示されていたが、作業毎の時間しか分からず、生産性を低下させる要因は見つけられなかつた。そこで、ビデオ撮影と画面キャプチャを組み合わせた行動分析の手法を考案、実施したところ、どのような行動が、どれぐらいの時間にわたつて行われているかの詳細が分かり、生産性が低い人特有の行動を明らかにする事ができた。

<キーワード>

自動テスト、テストファースト、ビデオ撮影、画面キャプチャ、行動分析

<想定する聴衆>

プログラム開発時を中心として、生産性向上を阻害している要因を特定したいと考えているマネージャー、開発者、プロセス改善推進者

<活動時期>

2015/11～2016/5

<活動状況>

- 着想の段階 (アイデア・構想の発表)
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他 ()

<発表内容>

1.背景

弊社では、Javaによるプログラム開発の品質向上のために自動テストを導入した。導入にあたっては、テストファーストで開発する手順(プロセス)を決定し、トレーニングも実施した。その結果、品質は良くなつたが、一部プログラマの生産性が下がつてしまつたため、自動テストの開発プロセスに問題があるのではないかと考えた。

プログラマには開発中の作業時間を5分単位で記録してもらつており、それによると自動テストのテスト仕様作成に時間がかかつてしまつた。生産性の問題を解消するために、これが事実であるかを見極める必要があつた。

2.改善したいこと

プログラマによる作業記録では作業毎の時間しか分からず、生産性を低下させる要因を見つけられなかつた。そこで今回、要因が特定できる分析手法を確立し、対策が実施できるようにすることにした。

3.改善策を導き出した経緯

あるキッチンメーカーでは、主婦が調理する様子を500時間以上にわたつて観察し、その結果を基にして顧客の9割が選択する定番商品を生み出した[1]。

我々のプログラム開発においても、開発中の様子を観察する事で、どのような行動が、どのくらいの時間にわたつて行われたかが明らかになり、生産性の高い人と低い人を比較する事で生産性に影響を及ぼす行動が分かると考えた。

また、以下のリスクを回避するために直接観察するのではなく、ビデオなどに記録してから分析する方法が良いと考えた。

- ・観察者が横にいることで、開発者が作業に集中できず普段とは違う行動をしてしまう可能性がある。

- ・リアルタイムで見ていると、観察者の思い込みや見落としにより真の行動が記録されない可能性がある。

4.改善策の内容

(1)測定方法

図1にプログラム開発プロセスと、今回の測定対象部分を示す。

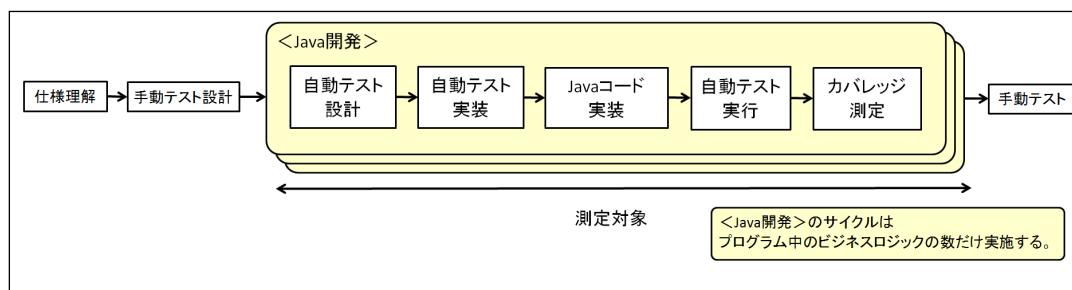


図1：プログラム開発プロセスと測定対象

測定対象としたプロセスで実施された全ての行動を、ビデオ撮影と画面キャプチャにより測定する(表1)。

表1：行動の種類と測定方法

行動の種類	測定方法	行動例
PC上で行う行動	画面キャプチャ	<ul style="list-style-type: none">・自動テストの実装・プログラムの問題点調査(コードリーディング) …等
PC外での行動	ビデオ撮影	<ul style="list-style-type: none">・紙資料の閲覧・紙にフローチャートを書いてのロジック検討・QA実施のための離席 …等

(2) 分析方法

図 2 に示す行動分析表を以下の手順で作成する。

手順 1. 縦軸を時刻、横軸を行動として、映像から行動を記録する。

手順 2. 表 2 に示す分類基準に従い、開発プロセスを分類する。

手順 3. 開発プロセス毎の時間、プロセス中の行動別時間を集計する(集計結果は 6 項)。

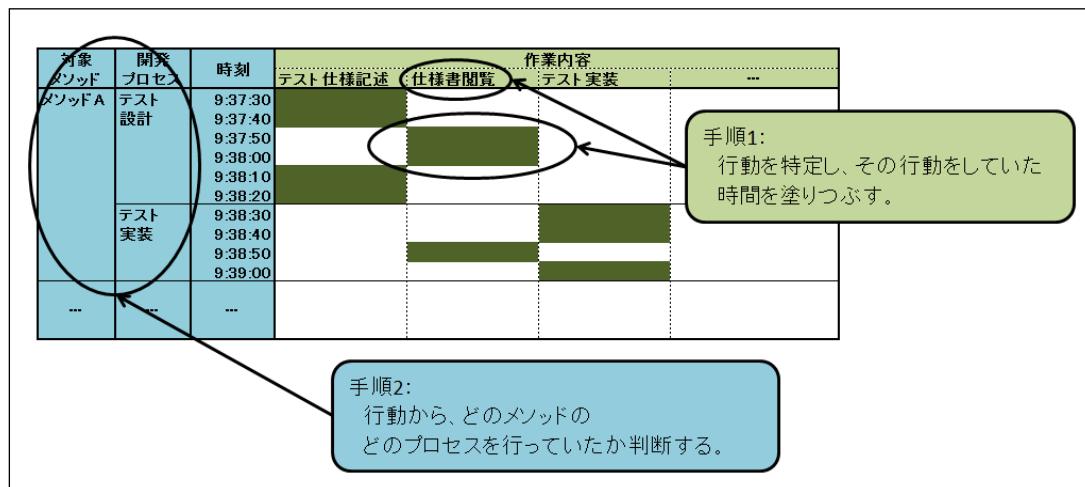


図 2 : 行動分析表

表 2 : 開発プロセス分類基準

開発プロセス	判定基準	
	開始基準	完了基準
自動テスト設計	自動テストソースコードに テスト仕様を記述し始めた時。	自動テスト実装が 開始された時。
自動テスト実装	自動テストソースコードに テストコードを記述し始めた時。	Java プラグイン実装が 開始された時。
Java コード実装	プログラムロジックを Java の ソースコードに記述し始めた時。	自動テスト実行が 開始された時。
自動テスト実行	1 回目の自動テストを 実行した時。	カバレッジ測定が 開始された時。
カバレッジ測定	1 回目のカバレッジ測定を 実行した時。	メソッド開発完了 基準を満たした時。

5. 改善策の実現方法

測定対象は次の通りである。

・対象プログラム : ある事務系登録画面(プログラム)の入力データをチェックする 3 メソッド(表 3)。

・被験者 : 初心者 A、熟練者 B の 2 名。

表 3 : 開発するメソッドの一覧

メソッド名	内容	規模(ライン数)
メソッド A	既にデータが登録済かのエラーチェック。	20 行
メソッド B	複数項目間のデータ整合性エラーチェックと、エラーメッセージの加工処理。	70 行
メソッド C	ヘッダ 1 件に対して明細 N 件が登録されているかのエラーチェック。	30 行
合計		120 行

6.改善による変化や効果

(1)分析結果

図3に両者のプロセス別時間を見た。熟練者Bの合計時間2.93(MH)に対して、初心者Aは6.17(MH)と2倍以上の差があった。プロセス毎の内訳を見てみると、初心者Aは全てのプロセスで約2倍の時間がかかっていた。

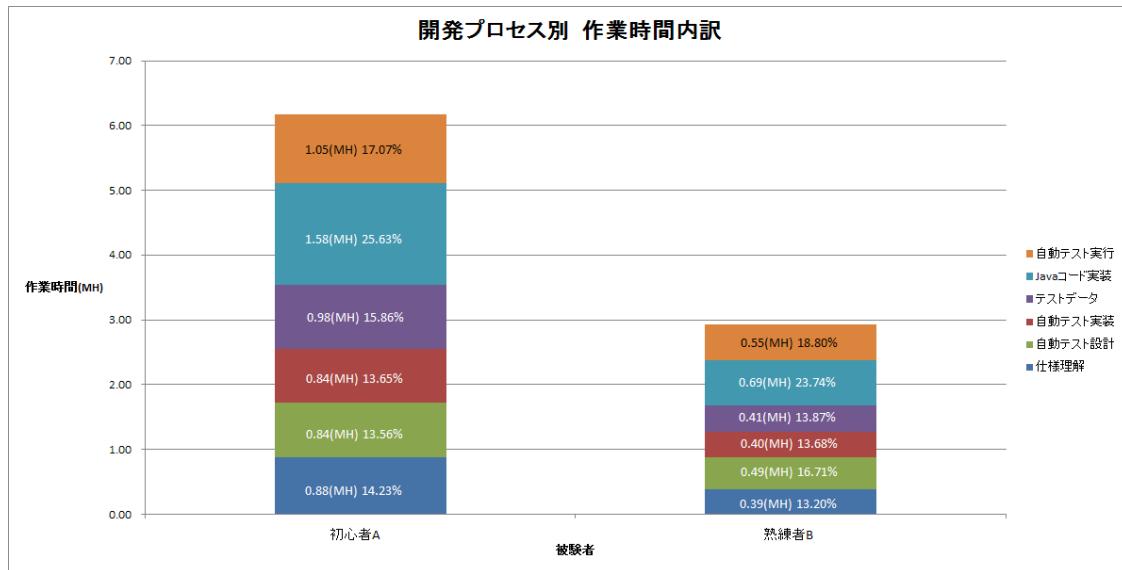


図3：開発プロセス別 作業時間内訳(左から初心者A、熟練者B)

次に、一番大きな差が見られたメソッドBの自動テスト実行プロセスにおける行動と自動テスト実行結果の関係を調査した。図4に色分けした行動と自動テスト実行結果を示す。

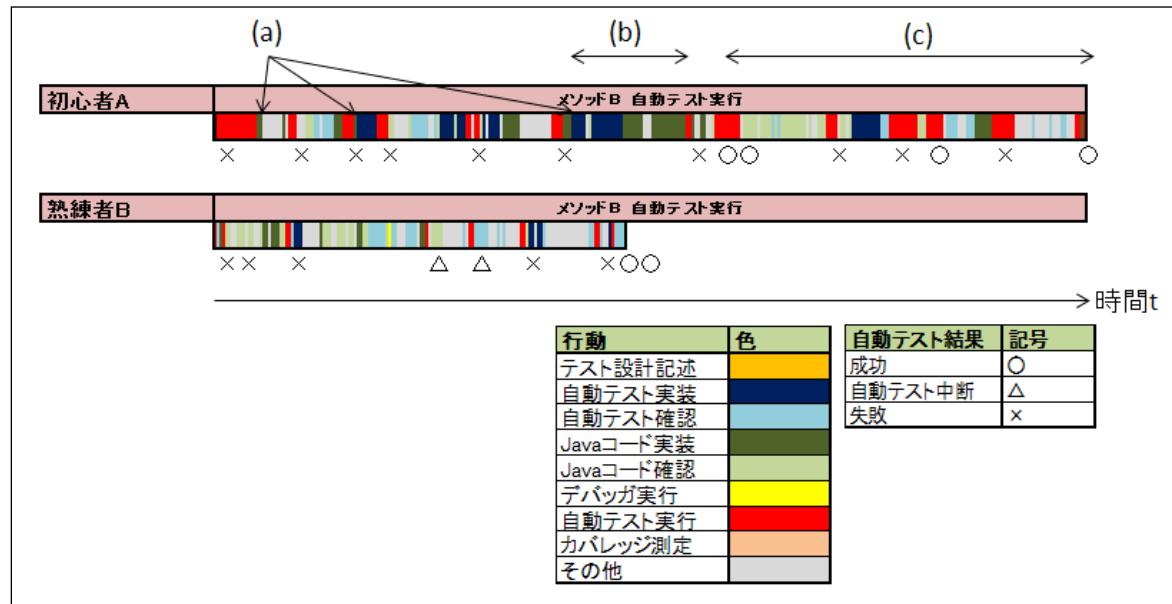


図4：メソッドB 自動テスト実行時の行動(上から初心者A、熟練者B)と、初心者A特有の行動

図4より、初心者Aと熟練者Bの行動を比較したところ、生産性が低いAには次のような特徴が見られた。

- ・自動テスト実行後のコード確認時間が短く、すぐに修正に着手している(図4の(a))。
- ・自動テストコード、Javaコード共に修正時間が長い(図4の(b))。
- ・自動テスト成功後もコードを修正している(図4の(c))。

このような特徴が生じた原因を次のように推測した。

- ・自動テストコードと Java コード共に品質が悪かったので、修正時間が長くなったのではないか。
- ・修正が場当たり的で、トライ & エラーを繰り返しているのではないか。
- ・自動テスト成功後も作業を行っているのは、仕様理解が不十分なままプログラム開発に着手したためではないか。

分析プロセスを改善した事により、生産性の低い人は各プロセスで少しずつ時間を要している事や、生産性が高い人には見られない行動を行っている事が明らかになった。また、これらの特徴を監視すべき新たな指標として提案する事ができた。

(2)分析プロセス改善の成果

ビデオ撮影などによる分析手法を確立したことにより、プログラム開発時にどのような行動が、どのくらいの時間にわたって行われているかが分かるようになった。

7.改善活動の妥当性確認

(1)分析コスト

分析時間は、開発時間のおよそ 2 倍(ビデオと画面キャプチャ)であった。頻繁には実施できないが、必要な時には行える程度の工数である。

(2)行動分析が適用できる状況

今回のようなビデオ撮影などによる行動分析は、改善すべき事が特定できないプロセスに適用できると考えられる。

A.参考文献

[1] 書籍「データサイエンティスト最前線」(2015 年)、日経 BP 社、20 ページ。

1C1 「開発者は不具合をどのように捉えているのか？不具合票のテキストマイニングをはじめる」 小島義也(エプソンアヴァシス)

<タイトル>

「開発者は不具合をどのように捉えているのか？」

<サブタイトル>

「不具合票のテキストマイニングをはじめる」

<発表者>

氏名（ふりがな）： 小島 義也（こじま よしや）

所属：エプソンアヴァシス株式会社

<要旨・主張したい点>

テキストマイニング技法のひとつであるネガポジ評価を用いて、ソフトウェア開発の不具合データから推測されるプロジェクトの感情評価を試みています。

また、ODC 分析を活用して、ネガポジ評価を実施することで、感情評価の傾向を絞り込むことを検討してみました。

<キーワード>

感情評価、ネガポジ評価、ZUNDA、テキストマイニング、不具合分析、ODC、市場影響

<想定する聴衆>

品質管理者、ソフトウェア設計者およびテストエンジニアの方々。

<活動時期>

2016/1～検証中

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

私たちの現実的な課題は、市場に出たソフトウェアの不具合が、ユーザーの不満足を引き起こし、製品そのものや企業イメージまで誤解を与えることです。

それは、私たちがつくった製品がユーザーにとって価値があるのか？といったことを問われていると思います。

仕様通りに作ったのに、できあがったものは外部の人が見るとちょっと違って見えることがあります。

そのギャップに答えるため、システム業界では製品の使い勝手の改善やユーザーの満足度の向上を図る手法が求められています。

しかし、これらを測るメトリクスは顧客満足度調査などでは登場しますが、ソフト開発のプロセスではなかなか使えるものにゆき当たりません。

そこで私たちは製品が市場に出る前の品質を、開発者がどのように捉えているのかを調べてみようと思いました。

まず、ソフトウェア製品のβ版リリース後によく見られる不具合には、仕様通りの動きだが使いにくい、ユーザーの目的と少しずれている、操作がまわりくどい、などという症例がでてきます。

これらの情報のひとつは、不具合票から取られています。

ここでは、これまでの開発フェーズでみられた素直なミスや勘違いに対する文章表現とは異なり、設計者の戸惑いや支持されなかつた仕様、ユーザー要求と実装の理論的ギャップ、不具合に対する代替的な仮説などが記載されることが多くなってきます。

これらはどのように評価されるべきなのでしょうか？

まず正しい仕様のあり方に至るプロセスの一部として捉えることができます。

そしてそれはまた、試行錯誤のプロセスもあります。

現実には、これらを客観的な評価へ方向付けしていくことが、プロジェクトマネージャの腕の見せ所であると思います。

そして、たいがいは「感情的」な部分は切り捨てて、そこに述べられているエッセンスを抽出し、このあとの改善につなげていきます。

しかし切り捨てられたこの「感情」のなかには何か、プロセス改善や製品開発の改善につながるヒントがあるのではないかと考えました。

仕様に適合すると評価された不具合票には肯定的な言葉（肯定的感情）が見られ、仕様に不適合とされたチケットには消極的な言葉（否定的感情）が見られることがあります。

それらの感情的な背景として、不確かと評価された仕様からは恐れや失望があり、確実と評価された仕様には喜びや期待があることから生じる文章表現なのだと思います。

これはある意味、「感情的」なことではないでしょうか？

そこには設計者としての誇りや恥があり、不具合と感情にはなにか関連性があるのではないかと考えた所以となります。

2.改善したいこと

では、いまのβ版のソフトウェア製品はどういう品質なのか、定性的に語るように、「感情的」に語るにはどうすればよいのかを考えてみました。

定量的なプロセス、プロダクト情報だけでは把握できない、感情的な情報を「見える化」するにはどうすればよいのか？

そこで、不具合の「ネガポジ評価」を試みてみました。

潜在的にチームが弱いと思っている要素をネガポジ評価から探れないものか？
そしてその要素は、プロセス改善や製品の改善への、ヒントとして使えないものか？

3.改善策を導き出した経緯

では、どのように不具合の「感情的な分析」を試みたかを説明していきます。

「感情的な分析」を行うためには、いくつか方法があると思います。
QC の方法論やなぜなぜ分析を行って、定量的な情報では把握できない、状況、気分、雰囲気といった定性的情報をより多く現場から引き出すことが考えられます。

今回はそうした方法の「現場から引き出す」ためのたたき台として、不具合票にかならずある、フリー入力項目に注目してみました。例えば「不具合詳細」や「補足」「備考」といった文章情報です。

これらは欠陥修正を行うプロセスにおいては重要な項目なのですが、その後はあまり顧みられない項目になります。
これらは定性的に扱うには面倒な情報であり、さらに定量化していくにはいろいろ手数を踏んで、クレンジングする必要があります。

そのため、今回は文章情報をそのまま扱う、テキストマイニングに注目してみました。そこで出会ったのが、「ネガポジ評価」です。
ネガポジ評価は、Twitter の感情解析による株価推定の研究で注目を集め、有名になりました。
これらの研究では Twitter のツイートに対してネガティブ評価とポジティブ評価を行い、インターネット上の言語データを数値化しています。

その結果、言語データの数値と株価変動の数値との間に何らかの相関関係があることを示唆した研究です。[1]

ネガポジ評価は、自然言語処理の技術を利用して、テキスト内の文章を数値化し、ネガティブな感情を示す単語とポジティブな感情を示す単語等を数値化して算出しています。

一般にネガティブとポジティブの 2 種類に絞って分析を行うものをネガポジ評価と呼んでいます。

今回は不具合票に含まれている情報から、意識がどれだけ「不足」していて（ネガティブ感情）、どれだけ「充足」している（ポジティブ感情）かとして捉えて、ソフトウェア品質に対する開発者の感情の傾向を見える化することを試行してみました。

4.改善策の内容

改善策の内容として、テキストマイニング技法のひとつであるネガポジ評価を用いて、アプリケーション開発の不具合データから推測される不具合の感情評価を試みています。

また、その傾向から今後のソフトウェアの欠陥予防が提案できないかを探ることを目的としています。

改善策の内容は大きく三つに分かれます。

①.不具合票の整理

・欠陥一覧を作成し、ODC による不具合分類を実施する。

②.欠陥一覧のネガポジ解析

- ・テキストマイニングによるネガポジ解析を実施する。

③.ネガポジ解析結果の評価

- ・結果データの傾向分析を実施する。

以下にその内容を示していきます。

①.不具合票の整理

一般的な不具合の分析は、不具合の発生状況から設計機能の弱い部分を洗い出していくアプローチになります。

これはレビューやテストで発見したバグは氷山の一角であると考えることからきています。

そして、同じ要因や原因、起因で似たようなミスが潜んでいるリスクを推測し、欠陥検出活動をことになります。

私たちはそのために、ODC（直行欠陥分類）分析を使って、不具合を様々な観点で分類しています。

「欠陥を発見する観点」

「その欠陥がお客様に与える影響の観点」

「修正した欠陥の種類の観点」

「欠陥を作り込んだ起因の観点」

などから、どんな不具合が発生しているのか、その傾向が見えてきます。

これらの不具合発生の傾向に基づいて、私たちは仮説を立てて適宜対策を打っています。

この不具合分類を当てはめたデータに対して、ネガポジ解析を実施することで、評価の観点を絞り込むようにしています。

②.欠陥一覧のネガポジ解析

テキストマイニングとは、文字列から構成された大量のテキストデータを解析して、その中から有益となる情報を取り出す技術のことです。

いまではテキストマイニング技術はオープンソースとして、様々なツールが公開されていて、簡単にパソコン上で試すことができます。

今回は不具合データをネガポジ解析するために、以下のような流れでデータ処理をしています。

②-1 ODC による分類別データ集計

②-2 ZUNDA による不具合データのスコア算出

②-3 R や Excel でのグラフ化

②-1 ODC による不具合分類は、人為的な原因を探る「欠陥起因」、欠陥がお客様に与える影響の観点としての「市場影響」分類でデータ仕分けを実施しました。

②-2 当初、一般公開されている有名な感情辞書を使用しました。

これは言語処理のための機械学習を研究している高村大也先生の「スピニモデルによる単語の感情極性抽出」の研究成果として公開されている、「単語感情極性対応表」のことです。[3]

その後、ネガポジの 2 値を含め、更に自然言語を解析できる「ZUNDA」を使用して、感情分析を試みました。[2]

ZUNDA は、「東北大学 乾・岡崎研究室で開発されたオープンソース 日本語拡張モダリティ解析器です。文中のイベント（動詞や形容詞など）に対して、その真偽判断（イベントが起こったかどうか）、仮想性（仮定の話かどうか）などを解析します。」

以下は、文章に存在する事象に対する、以下の 6 種類の項目をまとめて、拡張モダリティと呼ばれる解析項目です。

項目	項目概要
態度表明者	対象とする事象の成否の判断や、他者への働きかけや問い合わせをしている人物、もしくは、団体。多くの場合、この態度表明者は書き手である。
相対時	態度表明時から見た、対象事象の相対的な時間関係。過去・現在のことであるのか、それとも、未来のことであるのかを表す。
仮想	仮定された条件の有無。仮想世界の話であるのか、そうでないのかを表す。
態度	叙述、意志、働きかけ、問い合わせなどの伝達的態度。言語学における「表現類型のモダリティ」のカテゴリーに相当する。
真偽判断	態度表明者による対象事象の真偽判断。対象事象が成立か不成立かを、確信度とともに表す。
価値判断	態度表明者による対象事象の価値判断。対象事象の成立が望ましいことであるかどうかを表す。

ZUNDA は、解析した文に対し、上記の個々のラベルを設定しています。

ラベルの付与基準については、以下に詳しく説明されています。

「拡張モダリティタグ付与コーパス作成の作業基準 version 0.8β」より引用

項目	ラベル一覧
態度表明者	筆者
相対時	未来、非未来
仮想	条件、帰結、0
態度	叙述、意志、欲求、働きかけ-直接、働きかけ-間接、働きかけ-勧誘、欲求、問い合わせ
真偽判断	成立、不成立、不成立から成立、成立から不成立、高確率、低確率、低確率から高確率、高確率から低確率、0
価値判断	ポジティブ、ネガティブ、0

文章に記述されている情報は、単純な命題のみではなく、命題に対する情報発信者の主観的な態度も記述されています。これ

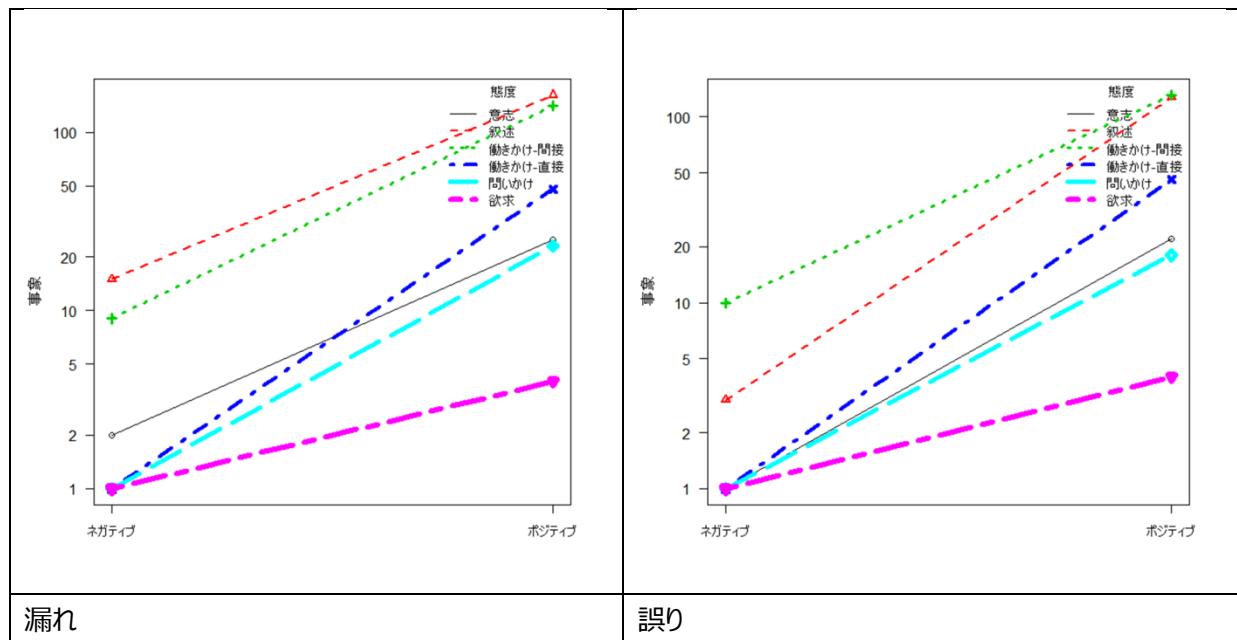
らの項目を使用して、不具合データの拡張モダリティ項目のスコア算出をしています。

②-3 スコア算出された結果を R や Excel で、ヒストグラムなどのグラフ化を行って、見える化していきます。

その結果から、不具合分類別の感情評価を分析しています。

以下では、検出不具合を「漏れ」と「誤り」にデータを分けて、感情評価した結果を見ています。

ODC 分類のひとつに、欠陥起因という項目があり、これで不具合を起こした人為的な原因の「漏れ」と「誤り」に分けて、ZUNDA でデータ解析をしています。



「漏れ」は、「叙述」が高くなっている、やはり設計漏れが原因の不具合であることを認識しての記載であったためと考えられます。またネガティブの「意志」も見られ、開発者が、ネガティブな自分自身の行為の実行／非実行を決定したことを表す事実もありました。

ネガティブの「意志」は、漏れではなかったという記載もあり、実際には仕様としてのグレーゾーンであったことを示していました。

「誤り」は、仕様変更と同傾向であり、「動きかけ-間接」が高くなっている、ミスした理由を記載しているためと考えられます。

5.改善策の実現方法

今回使用した不具合データは不具合管理システム（RedMine 上）に蓄積されており、不具合票として必要な項目のほか、テキストマイニングを行うメインデータとなる不具合の分析過程での解析結果などが登録されています。

そして、この不具合票を ODC 属性で分類したデータが、感情評価の対象となります。

当社では ODC による不具合分析の適用はいくつかのプロジェクトで実施してきています。

その一連の不具合分析活動のなかで、ODC 属性を利用した市場影響の分析を行っているフェーズもあります。

詳細は参考文献の資料を参照してください。[4]

不具合分析の流れは以下のようになります。

①.不具合票の整理

今回は、β版以降の不具合を対象としているため、ODC による不具合分類は、「その欠陥がお客様に与える影響の観点」としての「市場影響」分類でデータ仕分けを実施しました。

この「市場影響」は、以下のリスト項目から成っており、検出された不具合が、ユーザーからみた場合、どんな悪影響を与えてしまうかを推測し、設定する項目です。内容的には品質特性と似ています。

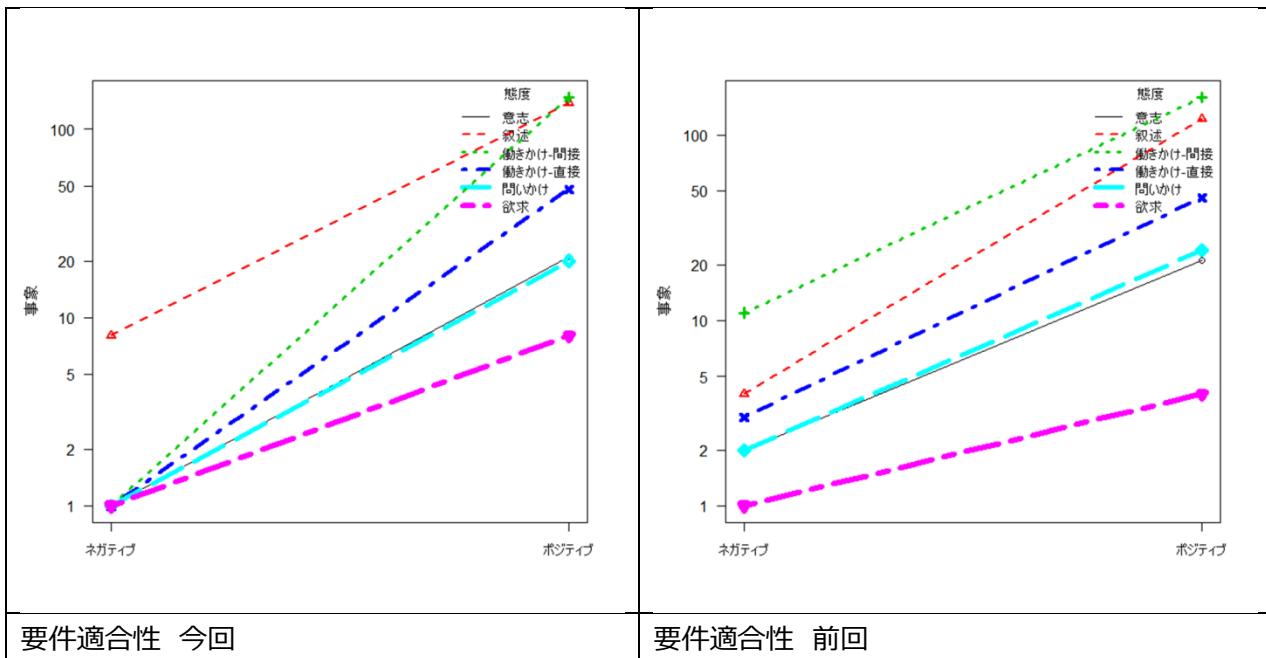
「市場影響」は 13 の観点が用意されており、この観点に不具合を当てはめて、分類していきます。[5]

以下の表では、[5]の森龍二氏の解説をもとに、観点を整理しています。（多少、文言を当社用に修正しています）

No	市場影響	Impact	説明
1	導入容易性	Installability	▶ソフトウェアを使える状態にするまでの容易性の観点。
2	完全性/セキュリティ	Integrity/Security	▶システム、プログラム、データなどの破壊・置き換え等への保護の観点。
3	処理性能	Performance	▶エンドユーザーが想定する処理性能、処理容量等の観点。
4	メンテナンス	Maintenance	▶ソフトウェアの保守性の観点。
5	有用性	Serviceability	▶処理の失敗を容易に、迅速に診断する能力の観点。
6	文書品質	Documentation	▶各種ドキュメントの正確性、利活用の容易さ等の観点。
7	移行性	Migration	▶新しいリリースに対して、既存のデータや業務への影響確認等が整備されているかの観点。
8	ユーザビリティ	Usability	▶ユーザーにとって、機能や説明書が理解しやすく、便利であるかの観点。
9	標準適合性	Standards	▶関連する規格・標準に準拠されているかの観点。
10	信頼性	Reliability	▶ソフトウェアが意図しない中断なしで、その目的の機能を実行できるかの観点。
11	要件適合性	Requirements	▶ユーザーの要求通りの機能、期待を満たしているかの観点。
12	アクセシビリティ	Accessibility	▶障害のある人に、情報や情報技術の利用が配慮されているかの観点。
13	可能性	Capability	▶エンドユーザーが既知の要求(業界標準や常識)を意図して実行した結果が、満足するものであるかの観点。

②.不具合の ZUNDA 解析とその評価

今回は市場影響を分類した不具合データごとに ZUNDA 解析を実施しています。



今回と比較すると、前回は「働きかけ-間接」のほかに、ネガティブな「働きかけ-直接」や「問い合わせ」が目立っています。例えば、β版の時点での比較をしてみれば、グラフの形から品質の傾向を予測することが可能ではないのか？但し、このデータでは開発者はあまり入れ替わっておらず、不具合票の記載レベルは安定しています。そして、今後この「働きかけ」の推移を見たり、比較することで、市場不具合に対する開発者の感情の傾向が見えてくるのではないか、と考えています。

6.改善による変化や効果

この検証結果による改善はまだこれからです。

また、感情評価のデータを継続したプロジェクトで評価結果を積み重ねることで、ポジティブ感情とネガティブ感情の変化を見ることが可能であるとも考えます。

同様にプロジェクトのメンバーの感情がどのように推移しているのかは、例えば日報やレビュー議事録からも読み取れるのではないかとも考えられます。

プロジェクトが動いている間に、感情評価の推移データを確認しながら開発プロセスをみていくことで、改善の気付きにできるのではないかと考えています。

また、そうした推移データの傾向から、今後のソフトウェア品質に対する欠陥予防が提案できないかを探ることを検討していきたいと考えています。

7.改善活動の妥当性確認

改善活動による妥当性確認もまだこれからです。

また、方法論の改善も必要だと考えています。

現在もまだ評価中であり、実際の否定的な欲求とは何か、ということも本質的な今後の検討課題であると考えています。

不具合を作り込むに至った要因はあるとしても、それを否定的に語ることにはどんな心性があるのか。

プロジェクトの効率主義、功利主義に因するものなのか、チームのメンバー間の問題なのか、こうしたリスク分析手法のひとつとして、

今後もテキストマイニングの応用が可能になっていけば良いと考えています。

A.参考文献

[1][ネガポジ解析による Web データと株価変動の相関関係評価]

佐藤謙太;小高知宏;黒岩丈介;白井治彦

<<http://ci.nii.ac.jp/naid/120005588275>>

[2][ZUNDA]

Zunda は東北大学 乾・岡崎研究室で開発されたオープンソース 日本語拡張モダリティ解析器です。

<<https://code.google.com/archive/p/zunda/>>

[3]高村大也, 乾孝司, 奥村学

[スピニモデルによる単語の感情極性抽出]

情報処理学会論文誌ジャーナル, Vol.47 No.02 pp. 627--637, 2006.

[4][設計プロセスの課題を ODC 分析で改善してみた ～設計欠陥の原因追求による、欠陥修正工数の削減～]

小島 義也 (エプソンアヴァシス株式会社)

<http://www.jaspic.org/events/sj/spi_japan_2015/>

[5]「ODC 入門」森 龍二 (株式会社エクサ)

<http://www.slideshare.net/mori_ryuji/odc-14036416>

<タイトル>

XDDP 導入／定着の失敗事例に対する病理学的処方箋

<サブタイトル>

～派生開発推進協議会 T22 研究会からの提案～

<発表者>

氏名 (ふりがな) : 葛西 孝弘 (かさい たかひろ)

所属 : 日本電気通信システム

<共同執筆者>

氏名 (ふりがな) : 斎藤 芳明 (さいとう よしあき)

所属 : 富士ゼロックス(株)

氏名 (ふりがな) : 梶本 和博 (かじもと かずひろ)

所属 : (株)エクスモーション

氏名 (ふりがな) : 石川 亘 (いしかわ わたる)

所属 : 日本システム技術(株)

氏名 (ふりがな) : 大段 智広 (おおだん ともひろ)

所属 : オムロンソフトウェア(株)

<要旨・主張したい点>

派生開発推進協議会(AFFORDD) T22 研究会では、派生開発の現場において実際に起きた XDDP 導入/定着の失敗事例を研究している。失敗事例にこそ成功に導く多くの教訓があると考えるからである。

本研究報告の特徴は、失敗原因の分析過程に失敗学が提唱する「原因まんだら」を導入したことで失敗事例を共通の原因でカテゴライズできたこと、そしてソフトウェア病理学的なアプローチから問題の発生を病気ととらえ、個々の病気に対する対応処置、さらに「原因まんだら」でカテゴライズした病気群に対する予防処置を処方箋の形でまとめた点にある。

本発表では、失敗事例から処方箋にまとめるまでの研究過程と、処方箋の活用について発表する。

<キーワード>

- (1) 派生開発(XDDP)
- (2) 導入／定着の失敗
- (3) 失敗学の原因まんだら
- (4) ソフトウェア病理学
- (5) 治療法／解決法
- (6) 予防法／対策法

<想定する聴衆>

XDDP の導入または定着において失敗経験のある組織の開発者、または、管理者。

これから XDDP に取り組もうとしている組織の開発者、または、管理者。

<活動時期>

2014 年 10 月～2016 年 5 月(継続中)

<活動状況>

- 着想の段階 (アイデア・構想の発表)
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他 ()

<発表内容>

1.背景

XDDP を導入したことで大きな期待が得られた「XDDP の成功事例」が多く聞かれる一方で、XDDP を導入したが期待した効果が得られなかつた「XDDP の導入の失敗事例」や、XDDP を繰り返すうちに効果が得られなくなつた「XDDP の定着の失敗事例」が散見される。

派生開発推進協議会(AFFORDD) T22 研究会では、失敗事例にも多くの教訓があると考え、派生開発の現場で実際にあった XDDP 導入/定着の失敗事例を研究し、失敗を繰り返さないための対策の立案に焦点をあてて活動している。

2.改善したいこと

失敗事例に対して、その原因を分析し解決策を求めるのは一般的な手段であり、発生している課題への対応には有効な手段である。我々も先ずは個々の失敗事例に対して、背景を含めて分析を行い、それぞれの解決法をまとめた。

しかし、個々の失敗事例に対する解決法を提示するだけでは、現場での活用面から考えて不十分だと考えた。失敗問題の本質の見え方が異なると別の問題として捉えてしまい、ここで提示した解決策が活用されないからである。

そこで本研究では、個々の事例を失敗原因によって分類し、かつカテゴライズすることで、問題の本質が見えやすくなると考えた。さらにカテゴライズした共通の問題に対する対策は、次の取り組みで問題を発生させないための予防策として提案できる。このように、本研究会では、単なる失敗事例の対策集ではなく、開発現場がより活用できる処方箋を提案することを活動の狙いとした。

3.改善策を導き出した経緯

問題の本質を捉えやすくするために、それぞれの失敗事例の原因を失敗学で知られている「原因まんだら」を使い、原因まんだらのカテゴリに合わせ分類を行つた。既知の原因まんだらの原因で分類することで、同一のカテゴリに分類された失敗事例から共通点が見いだせると考えたからである。

また、ソフトウェア病理学的な発想を導入し、問題の発生を病気が発症しているととらえ、個々の問題の解決法は治療法として、さらに原因まんだらで分類した同一のカテゴリの問題群に対しては共通に適用できる予防法として、処方箋の形でまとめることにした。

4.改善策の内容

個々の失敗事例に対して解決法(治療法)、失敗事例を原因まんだらでまとめたものに対して対策法(予防法)を提案している。解決法は、現在、XDDP の導入に取り組んでいるが躊躇している方々や組織への参考になるとものと考える。対策法(予防法)はこれから XDDP に取り組む方々や組織の事前準備の参考、すなわち XDDP の導入/定着に失敗しないための予防になると考える。

5.改善策の実現方法

下記のそれぞれの状況で、開発者と管理者の立場のそれぞれで改善策を実施できる。

① 炎上中のプロジェクトでの問題対応：治療

発生している問題と同様な症例を処方箋リストから探し、その治療法を処方する。

② XDDP の導入／定着に失敗したプロジェクトの次回への対策：再チャレンジ/リベンジ、再発防止

- ✓ 過去に XDDP の導入を試みたが、XDDP は使えない技術であると誤判断し導入をあきらめた組織や個人が、処方箋リストから真の原因に気づき、XDDP 導入の再チャレンジを決断するきっかけにする。
- ✓ プロジェクトで発生した問題と同様の症例を処方箋リストから探し、失敗原因を確認したうえで、次回に向けた再発防止の策定を計画段階で実施する。

- ③ これから XDDP に取り組むプロジェクトでのリスク回避：予防
回避したい失敗事例をリストアップし、優先順位を決め、優先順位に従い失敗事例の予防法を講じたうえで、導入計画を練る。

6. 改善による変化や効果

下記のそれぞれの状況で、開発者と管理者の立場のそれぞれで改善による変化や効果を期待できる。

- ① 炎上中のプロジェクトでの問題対応：治療
 - 問題解決までのスピードアップ
 - 問題に対する対策内容の質の向上
- ② XDDP の導入／定着に失敗したプロジェクトの次回への対策：再チャレンジ/リベンジ、再発防止
 - XDDP の導入をあきらめた組織や個人が、導入の再チャレンジ案を計画する
 - 次回プロジェクトにおける再発防止策の質と検討効率のアップ
- ③ これから XDDP に取り組むプロジェクトのリスク回避：予防
 - XDDP に取り組んだ際の失敗事例の発生確率を低くする

7. 改善活動の妥当性確認

改善案の提案までで、妥当性の確認はできていない。今後の活動課題である。

A. 参考文献

- [1] 清水 吉男,「派生開発」を成功させるプロセス改善の技術と極意,技術評論社,2007
- [2] 清水 吉男,「XDDP 導入失敗事例集」,派生開発推進協議会,2014
- [3] 畑村 洋太郎,失敗知識データベースの構造と表現（「失敗まんだら」解説）,独立行政法人科学技術振興機構 (JST) <http://www.sozogaku.com/fkd/inf/mandara.html>,2005
- [4] Capers Jones,ソフトウェア病理学,構造計画研究所,1995

<タイトル>

なぜなぜ分析はこれでうまくいく！～根本原因特定へのアプローチ～

<サブタイトル>

<発表者>

氏名（ふりがな）：林 真也（はやし しんや）

所属：住友電工情報システム株式会社

システムソリューション事業本部 第一システム部 第一開発グループ

<共同執筆者>

氏名（ふりがな）：

所属：

<要旨・主張したい点>

昨年実施したなぜなぜ分析のファシリテータ・トレーニングは、一定の成果を出したものの、根本原因に到達する率は低いという問題が見つかった。この根本原因へ到達できない原因を分析し、改善策を立てた。分析対象とする真因は事前準備で特定しておいてから実際の分析に入ることや、分析が滞ってしまったときには因果関係を可視化して問題箇所に気付かせること、等の改善策を分析のプロセス標準として文書化した。本年実施するファシリテータ・トレーニングではそれら改善策を活用し、根本原因への到達率が昨年から向上したかを判断・評価する。

<キーワード>

CMMI CAR 原因分析と解決 なぜなぜ分析 ファシリテータ 真因 根本原因

<想定する聴衆>

改善活動に係わる管理職

<活動時期>

2012年6月～現在

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

弊社のソフトウェア開発における組織目標は5年ごとに改定しており、目標達成のためにプロセス改善をする必要がある。それは進行中および完了したプロジェクトから多くの知見を集め、有効な改善策を見つけ、それを組織的に展開することが重要である。

原因分析をするための人材教育の1つとして、2015年4月から「なぜなぜ分析 ファシリテータ・トレーニング」を始めた。前回のSPI Japan2015ではそのトレーニング内容と成果を報告した。第1期のトレーニングでは分析できるファシリテータ6人を育成したという一定の成果を出したが、根本原因に到達した率が想定したものより低い結果となった。

2.改善したいこと

根本原因に到達できないということは、効果が高い対策を打てないことにつながり、ひいてはプロセス改善による組織目標の達成も困難になる。そのため、開発規模が大きなプロジェクト・小さなプロジェクトに係わらず、問題の本質的な原因（根本原因）をより確実に見つけられる技能を身につけさせることを、今回の改善目標にした。

3.改善策を導き出した経緯

第1期のトレーニング完了後に、トレーニング内容に対して振り返りを実施した。第1期のトレーニング記録はすべて残しており、その中からファシリテータとしての評価点は高いが、根本原因に到達できなかった回を題材にして、問題がどこにあったのかをWG内で分析した。

うまくいかなかったものから、分析の会議時間が短いもの、あるいは分析に入るまでに時間を使ってしまい十分な分析ができなかつたものが共通事項として見つかった。会議を1.5時間以上確保したにも係わらず、分析の時間が足りなくなつたものをさらに分析すると、分析の取り掛かりとなる真因（図1参照）を探すことによって時間を費やしてしまったことが分かった。

真因は技術的要因（図1参照）にあり、物的証拠という客観的な成果物を追えば見つかるはずである。分析の資料準備のときに、真因は特定可能だという認識に至った。

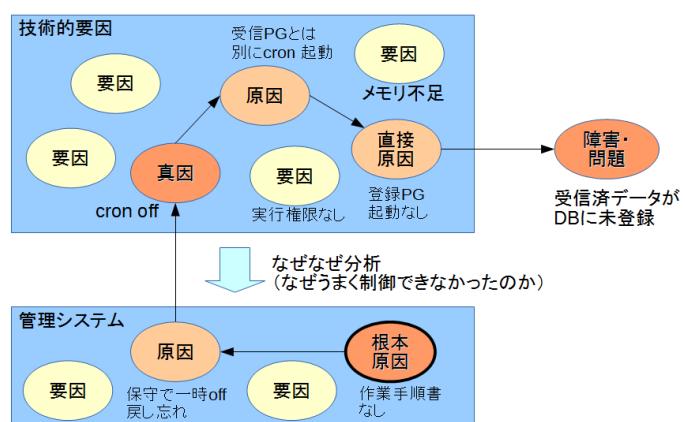


図1. 要因、原因、真因、根本原因

次に会議中に深掘りが進まずに沈黙が続いてしまい、分析を諦めて根本原因に到達しなかった事象が見つかった。停滞してしまった原因を分析すると、開発プロセスや手順等の観点をファシリテータが提示できていない、参加者に要因を気付かせる質問ができていない、事象の因果関係を漏れなく正しく書き出せていない、ことが上がった。何らかの方法でこの停滞を開拓する必要があった。

おおよそ共通した問題は上記の「事前に真因の特定ができていなかった」と「分析中に停滞が発生してしまった」であるが、分析会議に当 WG から参加するアドバイザーや、問題事象を分析する会議参加者に対し、問題が無かったかの確認を行った。すると、アドバイザーは深く介入せず分析会議を見守ることに徹したことがわかった。トレーニングの初期段階ではファシリテートが未熟なこともあいまって、分析をうまく進められず迷走したり、因果関係が取れていない分析をしたりする問題が見受けられた。どこまで介入してよいのか基準を設けていなかったために、根本原因に未達という結果となっていた。

第 1 期のトレーニングでは、ファシリテータはトレーニング開始前に事前教育を受けていたが、ファシリテータ以外の参加者は事前説明無しに分析会議へ出席していた。何をする会議であるのかをファシリテータが説明しなければならず、時間の有効活用を阻害する要因となっていた。参加者に事前知識をもつてもらう対策が必要である。

上記までの課題をまとめると、以下の 4 件である。これらを優先して解決すべき課題とした。

- ①分析前の準備プロセス定義
- ②分析停滞時の打開
- ③アドバイザーの介入度合い決定
- ④参加者への事前説明資料準備

4. 改善策の内容

3 項の①②③④それぞれで次の対策を打った。重要なポイントに絞って記載する。

①分析前の準備プロセス定義

準備で一番大事な事項は真因の特定である。分析会議で真因を探し始めたり、途中で真因の間違いに気付いたりして、限られる分析時間を有効に使えなくなる恐れをなくす効果を狙ったものである。

問題事象の報告を元にして集めたプログラムや仕様書、テストシナリオ、データ等から、時系列順で最初に発生した事象が真因であると、明確に定義した。特定した真因は「○○が△△であるべきところ××になっていた」といった表現で書き表す。さらに、これをなぜなぜ分析の最初のステートメントにすると手順を決めた。これにより何（○○）がどうなって（××）いたのが問題で、本来ならどうあるべき（△△）であったのかが端的に理解でき、分析を進める方向にブレがなくなり、かつ分析時間を有効に使えると考えての施策である。

②分析停滞時の打開

停滞を分析したときの原因それぞれについて対策を打った。

a. 多様な観点を提示

根本原因は管理システム（図 1 参照）にあり、それは目に見えないものである可能性もある。影響を与えるリソースをファシリテータが知らないと、参加者に観点提示できない。根本原因が潜んでいそうな要因を掘り出せることを狙い、ファシリテータの質問のレパートリーを増やすことを考えた。

考えられるリソース全てを文字情報として取り上げることは不可能であり、あまり事細かに書くとファシリテータがそれ以外の観点を考えなくなるので、気付きを与える程度の記述で観点提示することにした（図 2 参照）。

■作込要因

想定するサブプロセス：

外部設計、基本設計、PG設計、コーディング、テストシナリオ作成、本番化手順書作成、
変更設計、環境設定、議事作成、伝達、プロセス定義

なぜ、欠陥を作り込んだのか？ <i>IV:12J</i>		キーワード
1 [+]	【Management】	プロセス、組織、プロジェクト管理
2 [+]	【Material】	文書類、プログラム
3 [+]	【Man】	作業者
4 [+]	【Machine】	ハードウェア、ツール
5 [+]	【Media】	作業環境、コミュニケーション

■見逃要因

想定するサブプロセス：

ピアレビュー、CI、UT、UT検収、IT、ST、本番化リハーサル、影響調査、原因分析、
納品物検査、査閲、承認

なぜ、欠陥を見逃したのか？ <i>IV:12J</i>		キーワード
1 [+]	【Management】	プロセス、組織、プロジェクト管理
2 [+]	【Material】	文書類
3 [+]	【Man】	作業者
4 [+]	【Machine】	ハードウェア、ツール
5 [+]	【Media】	作業環境、コミュニケーション

図2. 特性要因図

b.行動から要因を引き出す

「〇〇した」「××だと思った」という当事者からの回答に対し、「なぜ？」と質問しても、的確な回答が得られそうにない。何かをしたり、何かを考えたりするときには、何らかの規則や手順、習慣が存在したはずである。行動自体に問題があったのか、行動の元になった規則・手順に問題があったのかを明確に切り分けられることを狙い、当事者の行動に対しファシリテータが行う質問の仕方を組み立てた（図3参照）。

ID	ファシリテータの質問と分析を進める箇所
1	【質問1】どのようなプロセスがありますか？
2	◆ CASE【回答1A】組織標準のプロセスが存在している
2.1	【質問2】そのプロセスを遵守しましたか？
2.2	◆ IF【回答2A】プロセスを遵守した 標準プロセスの問題点を分析
2.2.1	ELSE【回答2B】プロセスを遵守しなかった 守らなかった原因を分析
2.3	CASE【回答1B】プロジェクトのルールが存在している
2.3.1	【質問3】そのルールを遵守しましたか？
3	◆ IF【回答3A】ルールを遵守した ルールの問題点を分析
3.1	ELSE【回答3B】ルールを遵守しなかった 守らなかった原因を分析
3.2	CASE【回答1C】普段している習慣が存在している
3.2.1	【質問4】普段通りしましたか？
3.3	◆ IF【回答4A】いつも通り作業をした 実施者の習慣の問題点を分析
3.3.1	ELSE【回答4B】いつも通りにしなかった いつも通りやらなかった原因を分析
4	
4.1	
4.2	
4.2.1	
4.3	
4.3.1	

Key: if, else, case | loop, while, until, for, foreach

図3. プロセス遵守・不遵守 質問リスト

c.因果関係をモデル化

分析している事象に対し、ファシリテータと参加者のメンタルモデルが合っていないことがあり、質疑の会話だけだと空中戦をし

てしまい、時間を無駄に使うことになる。空中戦をさせないために事象の因果を書き表して、メンタルモデルを目で見える形にすれば、問題箇所が見つけやすくなるのではないか、と期待して具体策を練った。

考えた対策はホワイトボードを使い、参加者への質問と得られた回答を図としてその場で書き上げ、メンタルモデルを構築していく技法である（図4参照）。ファシリテーターが前に立って図を描くことで参加者の意識を当該事象に集中させ、その場で1つずつ因果を書き込むことで新しい気付きを誘発し、参加者全員に因果のイメージを共通化できることを狙っての対策である。その場・その時（ライブ：Live）に因果をモデル化（Modeling）することから、この手法をライブモデリング（略称：LM）と命名した。

なお、検討当初は真因から根本原因を遡りながら見つけるところで活躍すると考えていたが、当WGで実施方法の議論を深めしていくうちに、問題事象の発端から真因を特定する箇所でも同じ技法が使えることに気付いた。

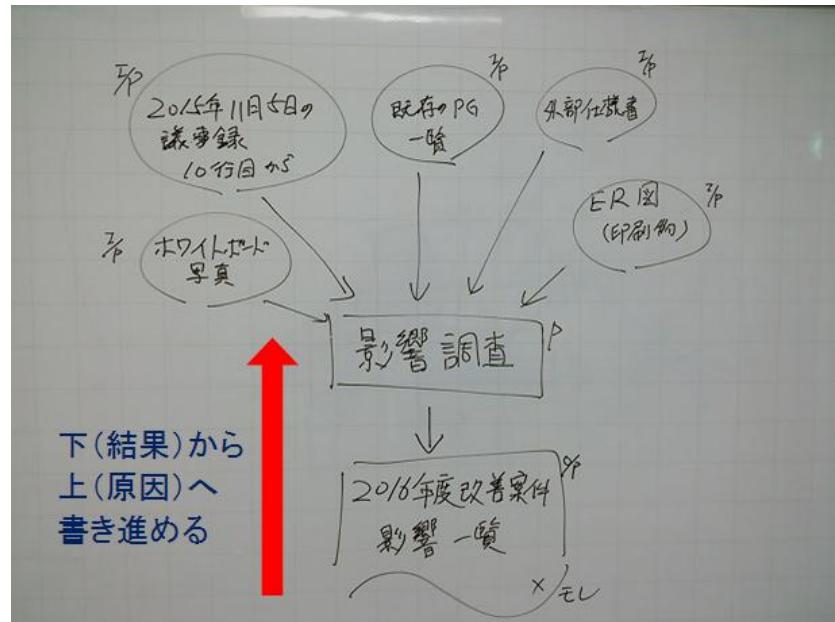


図4. ライブモデリングのサンプル

③アドバイザーの介入度合い決定

当WGで期待している一人前のファシリテーターに仕立てるため、分析中にアドバイザーは適宜介入しても良いとした。

④参加者への事前説明資料準備

なぜなぜ分析に協力および趣旨を理解してもらうため、用語の解説、分析の目的、悪い分析例を書いた資料を準備した。

5.改善策の実現方法

第2期のトレーニングは以下の方法で進めた。

- (1) 各部署から1名以上受講者を募る。
- (2) 当WGからはアドバイザーとして、各分析会議に1名参加する。
- (3) 分析対象は開発、保守のプロジェクトで実際に発生した事象とする。
- (4) 期間は3ヶ月とし、期間内に6回なぜなぜ分析をする。
- (5) 1回あたり分析会議を1.5時間、振返時間を0.5時間確保する。
- (6) 分析会議後の振り返りで定量評価、定性評価を行い、フィードバックシートを作成する。
- (7) 6回目終了時に、トレーニングに対する感想、意見を報告する。

おおよそ第1期と同内容であるが2点変えている。第1期の結果から根本原因に到達できていた時間が1.5時間であったので、(5)の時間配分とした。第2期では根本原因に到達しやすくする対策をいくつか打ったので、その効果を測るために(6)のフィ

ードバックシートに施策評価のための項目を設けた（図5参照）。

【施策評価】			評点		
根本原因特定に役立つ為に作成した標準類について診断をお願いします。各項目に評点(※)を記入してください。		※評点: 1できていない 2あまりできていない 3ままでできている 4できている 一評価対象外	本人	アドバイザー	平均点
分類	評価項目	本人	アドバイザー	平均点	
準備	51 今回、「原因分析の準備」標準を活用した				
	52 今回、「原因分析の準備」標準が有効であった				-
深掘	53 今回、「原因分析の深掘基準」標準を活用した				
	54 今回、「原因分析の深掘基準」標準が有効であった				-
実施	55 今回、「原因分析の実施手順」標準を活用した				
	56 今回、「原因分析の実施手順」標準が有効であった				-
問題特定 アプローチ	57 今回、「真因・根本原因へのアプローチ」標準を活用した				
	58 今回、「真因・根本原因へのアプローチ」標準が有効であった				-

図5. フィードバックシート（一部抜粋）

受講者は、分析テーマの選定、真因の特定、会議室の確保、参加者への連絡、なぜなぜ分析の実施、会議の振り返り、報告書の作成、という順で、テーマを変えながら6回分析を行う。なぜなぜ分析中に停滞したときには、4項②で検討した対策を駆使して停滞を打開する。

当WGからはアドバイザーとして分析会議に参加し、適宜指導を入れながら迷走を防ぐ。会議が終われば、受講者とともに振り返りを行い、会議を評価する。

参加者はなぜなぜ分析中は、ファシリテータからの質問に対し回答をして、ともに根本原因を突き止めていく。見つかった根本原因はプロジェクトに持ち帰り、対策の立案を行う。

6.改善による変化や効果

第2期のトレーニングはまだ途中段階であり、どの改善策がどの程度効いているかの分析はできていない。

なぜなぜ分析の場に出席したアドバイザーの所感で、以下の意見をいただいている。

- ・真因特定を準備段階で実施するようにしたので、分析がやりやすくなった。
- ・ライブモデリングで因果の事実関係を書き出せたので、問題が特定しやすくなった。
- ・会議進行中にアドバイザーが介入しても良いとしたので、それも効いた感じがする。

実施途中であるが、直近3回の根本原因の到達率を集計した。第1期は最終評価で37%の到達率であったのに対し、第2期は途中段階ではあるが52%の到達率と、1.4倍に増加している（表1参照）。

表1. 根本原因の到達率

期間	A: 受講者数	B: 根本原因到達数	C: 到達率 (B ÷ (A × 3))
第1期	9	10	37%
第2期	12	19	52%

根本原因到達数は終盤3回から集計
受講者数の3倍に当たる分析会議を実施している

本番の報告時点では第2期のトレーニングも完了し、最終到達率も確定し、改善策の効き具合も判明しているはずである。

7.改善活動の妥当性確認

今回の改善策を取り入れたなぜなぜ分析プロセスを実施することで、根本原因に到達する率が第1期の結果と比べて向上し

ている。目標として掲げた根本原因を見つける技能が向上したと判断する。

ファシリテータ向けの初期教育は受講者 1 名当たり 2 時間かかり、前回比較で 1 時間の増加があったが、根本原因の到達率が前回比較で 1.5 % 向上したことから考えると、教育コスト増に見合う価値はあると判断する。

以上から今回の改善策は妥当なものであると評価する。

A.参考文献

<タイトル>

ソフトウェア開発における品質管理の理論と実践

<サブタイトル>

欠陥件数の持つ代数的性質とその品質管理への応用

<発表者>

氏名 (ふりがな) : 小室 睦 (こむろむつみ)

所属 : 株式会社 プロセス分析ラボ

<共同執筆者>

氏名 (ふりがな) :

所属 :

<要旨・主張したい点>

工程別の欠陥摘出件数に対する理論的な枠組みを提案し、その品質管理、品質保証への応用を与える：

ソフトウェア開発での欠陥摘出数を工程別に並べた分布を考える。このような分布が（形状パラメータが一定の）ワイブル分布にしたがうと仮定すると、異なる工程間の欠陥摘出数の間に簡単な代数的関係式が成立する。この関係式を動機づけとして、上流工程での欠陥摘出数の変化が下流工程にどのような影響を及ぼすかを定量化した影響評価モデルを構築する。さらに、この影響評価モデルを変形することで上流工程での欠陥摘出数とプロジェクト情報から下流工程の残存欠陥数を予測する品質予測モデルを構築する。最後に、これらのモデルを実プロジェクトに適用した結果について報告する。

<キーワード>

品質保証、品質計画、品質予測、定量的モデル

<想定する聴衆>

SEPG、品質保証部門、プロジェクトマネージャ、プロジェクトリーダ、上級管理者

<活動時期>

2004-2015

<活動状況>

- 着想の段階（アイデア・構想の発表）
 - 変更を実施したが、結果はまだ明確ではない段階
 - 変更の結果が明確になっている段階
- #言葉がわかりません。「変更」って何？改善のこと？募集内容のどこにも「変更」という言葉はありません。
- その他 ()

<発表内容>

1. 動機

ソフトウェア開発がうまくいかない典型的な形態として、上流工程での品質作りこみが不十分なためにテスト工程以降で多大の工数を費やす場合がある 1). このような事態を防止するためには、上流工程での品質向上策、例えばレビューを徹底させればよいはずであるが、どの程度まで「徹底」させれば十分なのかその判断がうまくできないと、明確な根拠もないまま楽観的な見通しのもとにプロジェクトを進行させてしまい結局は失敗を繰り返すことになる。上流工程におけるレビュー実績データから、テスト工程での品質向上効果を示し、さらにはその後に実施するテスト工程での品質を予測するような定量的モデルがあれば、こういった事態を未然に防止でき、実用上重要である。

2. 工程別欠陥分布と信頼性工学手法

ウォーターフォール型のライフサイクルモデルを仮定する：一連の工程 p_i ($1 \leq i \leq n$) があり、特に、最後のいくつかはテストにあてられているとする。

上流工程でレビューを徹底すれば製品品質が向上するという定性的な観察を定量的に定式化することを考える。製品品質は下流工程、例えばシステムテスト工程での欠陥件数で測ることができる。上流工程での品質管理の徹底具合をどう定量化するかが問題となる。

2.1 指標定義

工程 p_i での（レビューまたはテストによる）欠陥摘出数を $f(p_i)$ であらわす。さらに、工程 p_i に対して、 $F(p_i)$ 、 $R(p_i)$ を以下のように定め、それぞれ累積前倒し率、残存欠陥率と呼ぶことにする。これらは信頼性工学における累積確率、残存確率（信頼度）の離散的類似である。 p_j がテスト工程の時、残存欠陥率 $R(p_j)$ は p_j 以降のテスト工程で摘出された欠陥の件数の全欠陥件数に対する比率であり、製品品質を表すものと考えられる。一方、上流工程 p_i に対する累積前倒し率 $F(p_i)$ は p_i までにどれだけの欠陥を摘出できたかという割合を示しており、レビュー等、上流工程での品質管理の徹底度合いを表しているものと解釈できる。

$$F(p_i) = \sum_{j \leq i} f(p_j) / \sum_{1 \leq j \leq n} f(p_j),$$
$$R(p_i) = \sum_{j \geq i} f(p_j) / \sum_{1 \leq j \leq n} f(p_j) = 1 - F(p_{i-1}).$$

2.2 ワイブル分布とべき乗減衰則

信頼性工学でよく用いられる分布にワイブル分布がある。これは確率密度関数が $f_{\alpha, \beta}(t) =$

$$\frac{\alpha}{\beta} \left(\frac{t}{\beta}\right)^{\alpha-1} \exp\left[-\left(\frac{t}{\beta}\right)^\alpha\right]$$

で与えられる分布である。

歴史的には、ワイブル分布は最弱リンクモデルに関連して導入され、システムの弱い部分から欠陥が発生していくと考えれば欠陥の時系列分布としてワイブル分布を期待するのは自然である。特に、ワイブル分布で

$$\alpha = 2$$

とした分布はレイリー分布と呼ばれ、ソフトウェアの欠陥件数や工数の分布に適用する研究 2)3) がある。ワイブル分布の残存確率関数は任意の異なる時刻 $t_1 < t_2$ に対して次のような等式を満たす。これをべき乗減衰則と呼ぶことにする。

$$R(t_2) = R(t_1)^c$$

ただし、 $c = c(t_1, t_2)$ は、に依存しない定数。

3. 二つのモデル

べき乗減衰則（の離散版）を仮定すると、異なる工程間の $F(p_i) = 1 - R(p_{i+1})$ と $R(p_j)$ の間の定量的関係を得ることができる。さらに、総欠陥数

$$\sum_{1 \leq j \leq n} f(p_j)$$

が開発規模 $size$ でほぼ決定されるという経験的事実を用いると次のような 2 つの仮説（モデル）を立てることができる。工程記号 ST でシステムテストを表す。 p_i はコーディング以前の工程を表すものとする。

効果説明モデル

$$\log(f(ST)) = -k_1 F(p_i) + A_1 \log(size) + (other factors) + \epsilon$$

品質予測モデル

$$\log(f(ST)) = -k_2 \sum_{j \leq i} p_j / size + A_2 \log(size) + (other factors) + \epsilon$$

ただし, $k_1, k_2 > 0$, A_1, A_2 は定数, $(other factors)$ は難易度, 開発経験などの要因の寄与, ϵ は誤差項を表す。効果説明モデルは p_i 以前の工程が ST 工程の欠陥件数にどう影響するかという品質向上効果を表すモデルであるが, 説明変数に $F(p_i)$ を含むため, 予測能力は持たない。品質予測モデルは, $F(p_i)$ を $size$ を用いた項で置き換える近似操作により, 予測(工程途中で入手可能なデータからの計算) を可能としたモデルである。近似のため精度は効果説明モデルより悪くなる。

4. 検証

前節で述べた 2 つの仮説（モデル）を実データにより検証した例を引用する 4). 使用したデータは 47 件のプロジェクトデータで, ポアソン分布を仮定した重回帰分析を適用した。 $F(p_i)$,

$$d(p_i) = \sum_{j \leq i} p_j / size$$

の係数, p 値, 回帰式の疑似決定係数の値を表に示す。

表 1 効果説明モデルの検証結果

工程 p_i 擬似決定係数 $F(p_i)$ の係数 p 値

機能設計	86.1%	-3.48	2.2 %
詳細設計	87.8%	-2.96	0.2 %
コーディング	85.9%	-2.46	3.0 %
単体テスト	91.9%	-5.57	0.1 % 未満

表 2 品質予測モデルの検証結果

工程 p_i 擬似決定係数 $F(p_i)$ の係数 p 値

詳細設計	85.4%	-0.069	0.9 %
コーディング	83.7%	-0.039	5.0 %
単体テスト	83.9%	-0.035	0.1 % 未満

5. 応用

効果説明モデルは, 品質計画を立案する際のシミュレーションや組織レベルでの改善効果の算定に用いていることができる。品質予測モデルは開発途中での品質見通しを立てるのに用いることができる。2 つのモデルを適用したプロジェクトで組合せテスト以降の欠陥密度が従来実績と比較して半減するなどの成果があがっている 4).

これ以外にも共同研究先で現在適用中の事例が複数あり, 発表スライドではこれらの事例についても説明する予定。これらの事例には、以下を含む:(1)途中工程に一括外注が入るため欠測値がある場合の適用例、(2) リリース後不具合件数の予測に用いた適用例。

6. 結論

ソフトウェア開発における工程別欠陥件数データの分布についてワイブル分布と同様の性質（べき乗減衰則）を仮定することで、上流工程の品質向上効果を示すモデルおよび工程途中での品質予測を実現するモデルを構築できることを示した。これらのモデルは実プロジェクトに適用され、テスト工程の欠陥密度を半減させるなどの成果をあげている。

A.参考文献

- 1) Humphrey, W. S.: *Winning with Software*, Pearson Education, Inc. (2002).
- 2) Putnam, L. H.: A General Empirical Solution to the Macro Software Sizing and Estimation Problem, *IEEE Transaction on Software*, Vol.4, No.4, pp.345-361 (1978).
- 3) John E. Gaffney, J.: On Predicting Software Related Performance of Large-Scale Systems, *Int. CMG Conference*, pp.20-23 (1984).
- 4) 小室 瞳：ソフトウェア開発における欠陥データ分布の統計的性質とその品質管理への応用、情報処理学会ワインターワークショップ 2016.

<タイトル>

マンスリー自己申告とクロスチェックの高速 SQA で赤字案件を削減

<サブタイトル>

無理なく自然と身に付くプロジェクトリーダの育成効果

<発表者>

氏名 (ふりがな) : 木戸 寛 (きど ひろし)

所属 : 株式会社大和コンピューター ソリューション統括本部 ソリューション 3 部

<共同執筆者>

氏名 (ふりがな) : 穴田 直也 (あなだ なおや)

所属 : 株式会社大和コンピューター ソリューション統括本部 コンサルティング部

<要旨・主張したい点>

SQA は、ドメイン、成果物、組織の文化、人員の気質等のバランスを保ちながら実施するのが有効である。当組織が、現場で試行錯誤して実践し効果を上げている SQA は、次のような特徴を持つ。プロジェクトの負荷状況、都合は一切考慮せず、毎月計画的に行う事で形骸化を無くしつつ効率的に、プロジェクトの軌道修正、日ごろの能動的な意識向上が見込める。また、複数のプロジェクトを同時かつ同じ場で監視する事で、プロジェクトリーダ間での参考意識、劣等感等を味わい、襟を正す効果も見込める。さらに SQA 実行担当として若手を当番制でアサインする事で、プロジェクト管理に必要な事項や手順を早期に学ぶ事が出来る。

<キーワード>

SQA、形骸化、効率的、プロジェクトリーダ育成、若手育成、プロジェクト管理、プロセス品質向上、標準化、プロジェクト状況確認

<想定する聴衆>

開発部門責任者、PMO、プロジェクトマネージャ、品質管理担当者、SQA 担当者

<活動時期>

2011 年 8 月～継続中

<活動状況>

- 着想の段階 (アイデア・構想の発表)
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他 ()

<発表内容>

1.背景

株式会社大和コンピューター ソリューション統括本部 ソリューション 3 部（以下、当組織）は、基幹系業務アプリケーションの受託開発を主に行っている。大手 SIer から開発を請け負っているほか、元請けに立つこともある。ISO9001 および CMMI に基づいた品質マネジメントシステムを運用し、納期が遅延するプロジェクトは一切なく、品質面でも顧客から高い評価を得ている。しかしコスト面では問題がなくはない。

2.改善したいこと

後戻りの作業が多かったり、作業が予定どおり進まなかつたりしても、要員の努力で品質を確保し納期を守る。そうするとコストに影響する。2008 年 7 月期から 2011 年 7 月期までの受託開発案件のうち、約 23% が赤字であった。赤字案件率を減らしたい。

3.改善策を導き出した経緯

以前は、他部門の人員が、本部レベルの組織標準に基づく評価基準を使用し、SQA を実施していた。他部門の人員ということで客観性は高かったが、当組織のプロジェクトの特徴や技術等にあまり精通していなかった。また本部レベルの組織標準に基づく評価基準は汎用的すぎて抽象度が高かった。これらの原因により、あまり深い指摘ができなかった。問題を見逃したり、プロジェクトにとっては的外れと思われる指摘があつたりした。プロジェクトにとっては効果が感じられず、メリットがなく、時間だけが取られた。

次に、SQA 担当者を他部門の人員ではなく、当組織の人員で実施することにした。そこそこ経験や知識のある別のプロジェクトマネージャやグループ長が SQA 担当者になった。他部門の人員よりは、プロジェクトの特徴や技術等を理解しているので、効果的な指摘ができた。しかし、次第に SQA 担当者のタイミングが合わないとか、SQA 計画を立てたけど実施しなかったとかで形骸化してしまった。

そこで、守るべき基準を明確にし、自律的でない人員にも実施可能な、効果のある SQA の形態が求められた。

4.改善策の内容

赤字案件およびコスト超過案件を分析すると以下の共通点が判明した。

- ・プロジェクトの計画が出来ておらず、作業役割、責任の所在が不明確。
- ・マスタスケジュールが存在せず、マイルストーンの共有が出来ておらず、顧客の都合により間延びする。
- ・作業の積上げと人員計画の両建てによるコスト及びスケジュールの妥当性が検証されていない。
- ・毎月、作業残の棚卸とリスクが出来てない。
- ・タスク（詳細スケジュール）の積上げが原価と一致しないまま進めている。
- ・納品成果物に含まれなければ、本来必要なものでも作成せずに進めてしまっている。またその必要性に関する意識も経験とスキルによりばらつきがある。
- ・品質が悪く戻り作業が多く余分なコストが掛かる。
- ・作業範囲外の事まで引き受けてしまい余分なコストが掛かる。

これらを防ぐために、本部レベルの原価明細や進捗管理表等の標準管理書式を改良したり、工数と日数の使い分け、テスト結果の残し方等のルールを決めたりして、当組織のための標準化を行った。また、標準の遵守状況を確認するための評価基準をチェックリストに含めた。

プロジェクト毎に SQA 担当者を割り当て、SQA 計画を策定し、それに従って SQA を実施するのはやめた。SQA の実施を次のように当組織内で統一した。月初めに、プロジェクトリーダが、チェックリストを使用し対象作業成果物を評価し、その評価結果を、EV、AC、CPI 等のプロジェクト監視の結果と併せて SQA & プロジェクト状況確認シートに記入する。プロジェクトマネージャがその内容を確認する。その後、グループ長以上、プロジェクトマネージャ、およびプロジェクトリーダが参加して、SQA & プロジェクト状

況確認ミーティングを開催する。その場で、各プロジェクトの状況を確認し、手順に則って遂行しているかを確認することでプロセス品質向上を図り、経験の浅いプロジェクトリーダには経験のあるグループ長やプロジェクトマネージャが指導することでプロジェクトリーダ育成を図り、トラブルを未然に防止するように必要に応じて軌道修正を行うようにした。

このように月に一度で一斉に SQA を実施するようにした。プロジェクトの進捗や遵守状況について管理層が相互にチェックできる。

また、記録係として若手社員を SQA & プロジェクト状況確認ミーティングに出席させることで、プロジェクト管理に必要な事を覚えさせ、若手育成を行うことも目的のひとつである。

5.改善策の実現方法

小規模短期間の案件に適用すると逆に予算を圧迫することは目に見えているので、500 万以上の案件を対象として上記の改善策を実施した。

SQA & プロジェクト状況確認ミーティングは月に 1 回、必要な時間は約 30 分。参加者はグループ長以上、各プロジェクトのプロジェクトマネージャおよびプロジェクトリーダ、ならびに記録係の若手社員 1 名（若手社員はローテーションで参加する）で 10 名に満たない。できるだけコストをかけず効率的に実施した。

6.改善による変化や効果

赤字案件率の期毎の推移は図 1 のとおりである。

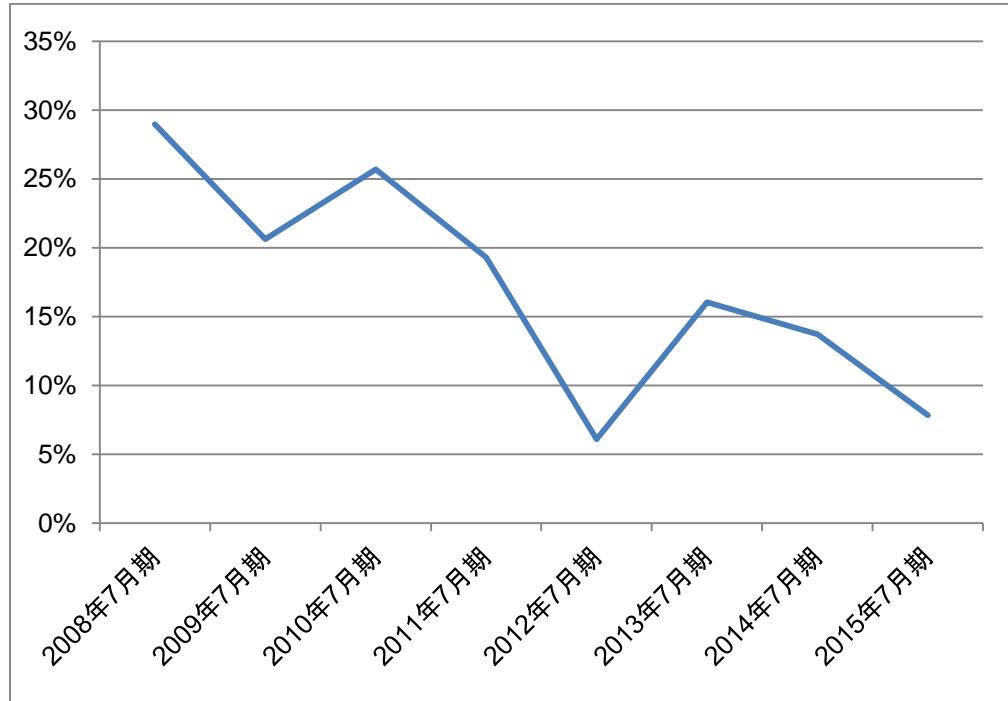


図 1 赤字案件率

2011 年 7 月期以前を改善前、2012 年 7 期以降を改善後とし表 1 のように 2 つの群に分ける。改善前 456 例と改善後 414 例を比較したところ、赤字案件数は改善前で 107 例(23.46%)、改善後で 47 例(11.35%)であった。 χ^2 乗検定を行うと、 $p < 0.0001$ であり、改善後は改善前よりも統計的に有意に赤字案件率が低かった。

表1 改善前/後と赤字/黒字の分割表

度数 行%	黒字	赤字	
改善後	367 88.65	47 11.35	414
改善前	349 76.54	107 23.46	456
	716	154	870

7.改善活動の妥当性確認

赤字案件はまだなくなつてはいないが、赤字案件率は減少した。

目に見えて品質と生産性が高まってきた。SQA & プロジェクト状況確認ミーティングへの若手社員の参加により、皆が何をしないといけないかが頭に入りやすくなったと思われる。

しかし、約 26%の案件が、赤字になつてないが当初予定した粗利を達成していないという結果であった。当初予定した粗利を達成する案件を増やしていくことが残存課題である。

A.参考文献

<タイトル>

失敗に学ぶ。再発防止策の立案・実行から定着するまでの仕組み作り

<サブタイトル>

<発表者>

氏名（ふりがな）：井之内 博夫（いのうち ひろお）

所属：オムロンソフトウェア株式会社 ベースソリューション事業部 品質管理グループ

<共同執筆者>

氏名（ふりがな）：

所属：

<要旨・主張したい点>

障害が発生すれば、"暫定対応（復旧）" → 障害原因の分析 → "恒久対応"に取組み、"本質的要因の分析" → "再発防止策"と続く。

障害対応は、どちらかと言うとネガティブなイメージで出発するものであるが、そこから得られる貴重な教訓を組織に広げ、定着させることができれば、個人の成長と組織の発展につながるというポジティブなものに終着させることができる。

このような障害対応から再発防止策までをプロジェクト管理ツールを用いた管理手法を確立することで、一連の管理をスムーズに行い、見える化を進め、品質問題の再発を抑止するとともに、失敗に学び、得られた教訓を伝承し続ける組織文化を創り上げていく活動に取組んだ。

<キーワード>

障害対応管理、再発防止策実行管理、見える化、共有、横展開、浸透、定着

<想定する聴衆>

品質管理・品質保証部門の方、設計・開発部門の方

<活動時期>

2015年4月頃から、現在も継続中

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

本発表の対象となる活動に取組んだ主体は、プラットフォーム開発チームとそのメンバ、品質管理グループとそのメンバである。

本取組みの経緯とそのきっかけを、次に述べる。

弊社では、新たにオープン系システムによる Web サービスのプラットフォーム開発を担うことになり、弊社の複数の事業部からメンバが集まり、プラットフォーム開発チームが発足した。

しかしながら、事業ドメインとしては新しい領域であり、またプラットフォーム開発の経験が豊富なメンバが揃っている訳ではなく、いくつかの品質問題を引き起こしてしまった。

これまでにも再発防止策に取組み、様々な方策や手段を尽くしてきたが、その全てが定着し、効果として表れてきたわけではない。

日々、新しいものを生み出していく中、新しい技術の修得、新しい発想が求められる一方で、過去の失敗から得られる教訓も大事であり、一人一人が得られた教訓を確実に身に着け、個人ではなく組織の資産としていくための仕組み作りが必要となった。

2.改善したいこと

「障害発生 → 暫定対応（復旧） → 障害原因の分析 → 恒久対応
→ 本質的要因の分析 → 再発防止策」

障害対応から、再発防止策を立案し、取組むまでの流れは定着している。

課会の議題に障害報告を取り上げるなどして開発チーム内で共有はしていても、実質的には障害を対応した担当者の資産に留まっていることは否めず、開発チーム内の他のメンバに横展開し、浸透できていないことがある。

これまでにも、作業手順、チェックリストの作成や整備は、繰り返し行ってきたが、時間の経過とともに形骸化してしまうことがあった。

また、障害の混入を防止するのか、流出を防止するのか、立案した再発防止策が何を目論み、何を意図しているのかが明確になっていないこともあった。

その結果、類似する障害が再発してしまうことがあり、信頼が低下するだけでなく、開発チームメンバのモチベーションの低下を引き起こしてしまうことにもなる。

障害対応から、再発防止策の立案・実行までの流れを見る化し、効果のある改善策が実行され、それが定着していくまでを共有しながら、P D C Aを回していく仕組み作りに取り組んだ。

3.改善策を導き出した経緯

現状の障害対応から再発防止策までの流れを分析すると、以下のようことが分かった。

- ① 再発防止策の立案、計画は、そのほとんどが実行策を仕上げるまでであり、実行策を横展開し、浸透させていく方法や手段については、十分な検討がされていない。そのため、再発防止策が開発チームメンバーに必ずしも浸透していない。
- ② 再発防止策の管理は実行策まであり、再発防止策が浸透するまでの管理がされていない。また、定着したかどうかの判断もできていない。
- ③ 立案された再発防止策が、障害の混入防止なのか流出防止なのか、再発防止策で狙いたい効果が絞り切れていない。

4.改善策の内容

①再発防止策の管理は、再発防止策が浸透し、定着するまでを管理対象とする。

そのために、再発防止策の計画には、完了基準、定着基準という2つの項目を新たに設けた。

再発防止策を立案する際には、もう一步踏み込み、浸透策や定着するまでの方法を立案検討すること、更に、定着したかどうかを判断する基準も立案・検討してもらうようにした。

完了基準 … 再発防止策の実施（手順書作成やツール作成などの具体策）が完了した状態を挙げる。

再発防止策の目的が達成できる対策が実施できた状態を挙げる。

定着基準 … 関係各位・各所において、再発防止策が浸透し、定着したと判断できる状態を挙げる。

再発防止策の目的が達成されている状態を挙げる。

②“障害対応の管理（障害発生から恒久対応まで）”と“再発防止策の管理”に分けて管理する。

障害1件に対し、必ずしも再発防止策1件というわけではない。

障害：再発防止策 = 1 : 1、 n : 1、 1 : m、 n : m

障害の内容、分析結果から、導き出される再発防止策は様々であり、不要の場合もある。

③再発防止策を立案する段階で、混入防止／流出防止を分類する。

明確に分類できない場合もあるが、立案する段階における検討課題とした。

混入防止 … 設計やコーディングの段階で、類似障害の混入が防止できるか。

流出防止 … レビューやテストなどの検証行為を通じて、類似障害の流出が防止できるか。

5.改善策の実現方法

①インフラの活用

改善策を実現するインフラとして、Backlogを利用することにした。

Backlogは、既にプロジェクト管理や障害対応管理に運用していて、インフラとして定着していたからである。

障害対応管理では、障害発生時のチケット発行から恒久対応までを管理することが定着していたので、恒久対応に続き、本質的要因の分析、再発防止策の立案というタスクを追加し、この後に繋がっていく“再発防止策実行管理”を新たにBacklogのプロジェクトに設けた。

こうすることによって、再発防止策の実行と定着までを独立した1つのプロジェクトとして、管理できるようにした。

②インフラを利用した改善策の実現

Backlogに再発防止策実行管理を新設した際のポイントを以下に挙げる。

・再発防止策実行管理のチケット発行時には、混入防止か流出防止に分類するようにした。

・再発防止策の実施完了の基準と定着判断の基準を明記するようにした。

・再発防止策の対応責任者に任命された者が、実施完了と定着までを管理することにした。

③改善策を実現するための支援・促進

再発防止策の管理対象を定着するまでにしたことは、開発チームメンバに戸惑いをもたらさせていた。

従来から、障害対応は恒久対応をもって一旦完了とするため、その後の再発防止策は緊急性が下がる傾向にあったが、更に、浸透策、定着基準の立案が加わったため、再発防止策の立案の遅れ、立案されても実行の遅れが生じた。

これらに対しては、品質管理グループが主導し、再発防止策の対応状況を開発チームメンバと共有していった。

・再発防止策の“立案”、“完了”、“定着”的件数を集計し、全体の対応状況を定量化した。

・担当者毎の対応状況（担当件数、ステータス）を見える化した。

・棚卸しを行い、遅れに対して手を打つ機会を設けた。

また、その際の留意点を以下に挙げる。

- ・単なる催促は行わないこと。
- ・その再発防止策が、どのような効果をもたらすのか、次のどの作業までに必要なのか、再認識してもらう。
- ・遅れを生じさせる原因、困りごとなどを発信してもらう。

6.改善による変化や効果

- ✧ 再発防止策を浸透、定着させるまでの方法は様々で、定着したかどうかの判断基準も様々であるが、定着基準を設けることで、実行だけでなく、定着するまでを管理するプロセスが確立できた。
- ✧ 再発防止策を通じて、どのような技術、作業ノウハウが組織に浸透しているのか見える化することができた。
- ✧ 混入防止、流出防止を分類するようにしたことにより、再発防止策が何を目論み、意図しているのかが、立案時に検討され、明確に分類することができるようになった。

7.改善活動の妥当性確認

- ✧ Backlog のチケットによって、再発防止策が開発・設計や障害対応などと同じように管理されるようになり、再発防止策が障害対応の延長線上にあるのではなく、独立したものとして取組み、定着するまでを管理することができるようになった。
- ✧ 講会の議題に障害報告を取り上げて共有したり、チェックリストに頼るだけでなく、技術知識、作業ノウハウを共有する機会が設けられ、再発防止策は定着するまでを管理するものという意識が芽生えてきた。
- ✧ 再発防止策を1つのプロジェクトとして管理していくために、プロジェクト管理ツールとして既に定着していた Backlog を活用したことも効果的であった。人の行動を変えることは容易なことではなく、既にあるもの、慣れたものをひと工夫することは、改善策を加速、促進していく上での教訓であると言える。

今後の課題

- ・再発防止策の実行と定着までは見える化できたが、その効果を確認する仕組みを確立していく。
- ・再発防止策の取組みそのものの効率化と更なるスピード感が求められる。

A.参考文献

- 「失敗学のすすめ」畠村洋太郎（著）
- 「失敗学（図解雑学）」畠村洋太郎（著）

（活用しているインフラ）

Backlog 株式会社ヌーラボが提供するプロジェクト管理ツール

以上

2A3 「危険予知トレーニングによる障害の未然防止」 南波貴紀(インテック)

<タイトル>

危険予知トレーニングによる障害の未然防止

<サブタイトル>

障害発生件数の激減をめざして

<発表者>

氏名（ふりがな）：南波 貴紀（なんば たかのり）

所属：株式会社インテック 行政システム事業本部 東北行政システム部

<共同執筆者>

氏名（ふりがな）：中村 勝（なかむら まさる）

所属：株式会社インテック 行政システム事業本部 東北行政システム部

<要旨・主張したい点>

当社では過去実際に発生した障害事例をベースに、障害発生の危険予知能力を高めるためのトレーニングを行なうことで、障害の未然防止活動を行った。これまでの品質改善対策に加え、自ら気づくことの大切さを従事するメンバー全員が認識できるトレーニングとして現在も継続実施している。

本発表では、当部が取り組んだ危険予知トレーニングの方法や運用、マネジメント、実施後の効果について紹介する。

<キーワード>

障害対策、未然防止、危険予知、品質改善、リスクマネジメント、トレーニング、ヒューマンエラー、KYT、疑似体験

<想定する聴衆>

適用業務システムの運用・保守プロセスに関わるアプリケーションシステムエンジニア（SE）やマネージャの方。

ITSS の職種では、主にアプリケーションスペシャリスト、IT スペシャリスト、IT サービスマネジメント、プロジェクトマネジメントの5つ。

<活動時期>

2014年9月から2015年9月まで（それ以降は定常業務として継続実施中）

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1. 背景

適用業務システムの運用保守プロセスにおいては、法律・条例の変更にともなう改修やお客さまからの機能改善要求などによる改修で、システムのバージョンアップがたびたび行われる。お客さまに本番リリースする前に十分なテスト・検証を行っているにもかかわらず障害が発生してしまう現実があり、大きなものは社会不安に結びつくこともある。当部においても例外ではなく、2013年から2014年までの一年間で、月に2件から多いときは十数件の障害が発生していた。

これまでの主な品質改善対策は下記三つであった。

- ① 障害の「なぜなぜ分析」 …… 根本原因に対する対策立案
- ② 品質ミーティングの実施 …… チェックリスト強化やルール強化や確認体制強化などの対策の横展開
- ③ 朝礼での「SE 十か条」の復唱 … SE¹のヒューマンエラーに関する教訓の日常確認

しかし、障害発生件数は思うように減らない状況下、お客さまにご迷惑をかけることで担当の営業やSEたちも疲弊気味の中、部門リーダーたちはどうアクションすればよいか苦しんだ。

さらなる対策を立案する必要があった。

2. 改善したいこと

課題は大きく二つある。

第一の課題は、「障害発生件数を減らし、お客さまにご迷惑をかけないこと」であり、月に2件から11件発生していたものをゼロ件にすることである。

第二の課題は、「納品物に対するSEの品質意識の向上」であり、これまで以上に品質の保証に貪欲になってほしく、その結果、出荷判定会議の内容が充実し、「気づき」が多くなることである。

3. 改善策を導き出した経緯

ある管理者が、当社で全社的に実施していた情報セキュリティ対策の一部である危険予知トレーニング（KYT）²を障害予防に役立てられないかと考えた。

しかし、KYTの方法論を障害対策に適用するには、考慮しなければいけないことが二つあった。

一つは、「長期的に効果が持続すること」であり、もう一つは「100%でないとダメ（関係者全員がすべて同レベルになること）」である。

効果があり、その効果が持続する保証など何処にもない。しかし、経験したことのない障害が発生する状況を設問とし、担当SEに問いかけて考えさせ、経験者と共に考え方を修正・共有することは未然防止に必ず役立つと管理者たちは確信した。

まずはやってみようとのことで合意形成できた。

「100%でないとダメ」については、情報セキュリティ事故と同様で、ひとりでも理解度が低いメンバーがいるとそのメンバーの関わった作業が障害誘発に結びついてしまうからである。これについては、個人演習の後に、リーダーを含めたグループディスカッションで個人差を解消することとした。

ところが、肝心のKYTの設問、つまり例題をどう作るかがひとつの思案どころになった。自部門の障害の傾向をとらえ、経験の浅いメンバーが疑似体験でき、正解は複数あり、適切な考え方や行動に結びつけることができなくてはならない。これについては、経験豊富なリーダーやベテランSEが実際に起きた事例をベースに作ることにした。

出荷品質の改善は緊急改善課題であり、部門長からも、「まずはやってみよ」と言うことで理解を得た。

4. 改善策の内容

これまでの品質改善策に加えて、危険予知トレーニング（KYT）による障害の未然防止活動を行うことになった。

以下に、トレーニングの概要と内容を説明する。

¹ SE:システムエンジニアの略（主に適用業務アプリケーションの改修・テスト・据付に携わる技術者）

² KYT:Kiken Yochi Training の略（中央労働災害防止協会が提唱）

² KYT:Kiken Yochi Training の略（中央労働災害防止協会が提唱）

4.1 トレーニング概要（5W1H）

- ① Why（目的） : トレーニングの目的は障害発生を未然防止すること。
- ② What（課題） : 障害の疑似体験で、経験の浅いSEが、適切な考え方・行動ができるようになること。
- ③ Where（対象範囲） : 事例は「重大な障害」と「軽微な障害でも数が多いもの」
- ④ Who（実現体制） : 企画準備は管理者とマネージャ、受講者はすべてのSE・リーダー
- ⑤ When（スケジュール） : 4ヶ月に一回実施（全員が時間の取れる時期は年3回のため）
- ⑥ How（実現方法） : 事前準備、実施（個人演習とグループディスカッション）、事後フォロー

4.2 トレーニング内容

【事前準備】（期間は1ヶ月）

- (1) 開催期間とマイルストーン、受講者グループを決める。

一回あたりの実施期間は約2ヶ月である。マイルストーンは、キックオフ・演習問題レビュー完了・個人演習完了・グループディスカッション実施日・結果のフィードバック完了の5つある。

受講者グループは、約6人を1グループとし、リスクマネジメント力に長けている上級SEは分散させる。

- (2) 演習問題を作成し、事前配布する。

個人演習、グループディスカッションともに「危険予知トレーニングワークシート」を利用する。ワークシートには、「A：問題と状況設定部分」、「B：危険要因と想定事故記入部分」、「C：対応策記入部分」がある。

リーダーとマネージャは、「A：問題と状況設定部分」を作成し、受講者に事前配布する。

【実施1（個人演習）】（期間は1週間）

- (3) 受講者が演習を実施する。

受講者は、「A：問題と状況設定部分」を読み、自分なりに「B：危険要因と想定事故記入部分」を書き、さらに、「C：対応策記入部分」を個人演習として行ってから、グループディスカッションに望むことになる。

ここでのポイントは、「C：対応策記入部分」にある。この対応策は、自分だったらどうするか、または、日ごろから自分は何をしているかを、書かなければならない。つまり、第三者ではなく、当事者としてリスクに向き合う姿勢が培われることになる。

【実施2（グループディスカッション）】（期間は3日間）

- (4) グループで意見交換し、まとめた結果を発表する。

1グループ6人で3グループが集合し、一日に2時間かけて実施する。グループで「B：危険要因と想定事故記入部分」と「C：対応策記入部分」について意見交換し、グループとして模造紙にまとめて発表する。一日に3グループずつ3日間をかけて、合計9グループ行った。一日あたり3グループずつに分散することで、アドバイザー兼講評者のマネージャの目が各グループに行き届く、また、全グループが発表できるメリットがある。

考えさせることが大切なので、グループでの意見交換においてベテランSEはなるべく発言せず、若手や未経験者の意見を引出すよう心がけさせた。

【事後フォロー】（期間は0.5ヶ月）

- (5) アンケートと結果の共有。

グループディスカッションが終った後、今後の改善データとするために、アンケートにて理解度・実用性・時間配分などを確かめる。また、個人演習の結果やグループディスカッションの結果をまとめ、全員で共有する。

5.改善策の実現方法

危険予知トレーニングは部門トップの強い意思の元に、全員で早急に実施することになった。まずは半年間続けてみて効果があればさらに続けようということで、実施内容や運用体制をルール化した。

・苦労した点

演習問題を作るのが大変であった。実例をそのままでは生々しくて使えないこと、必ずしも目的に合ってはいなくて議論が拡散してしまう可能性もあるので、ある程度スリムにして演習問題（前述の「A：問題と状況設定部分」）にする必要があった。ま

た実例の当事者を傷つけないように配慮も必要である。目的やねらいについては、マネージャ以上の合意を得ることや、正解は一つでないにしても、うまく誘導できるようにつくらねばならない点に注意してレビューを重ね、シナリオを想定することが必要であった。

また、最初のころは、グループディスカッションで時間が足りなくなることがあり、時間配分に苦労した。これについては、自分でできる個人演習を終えてから集合することにした。これにより、グループディスカッション前に、アドバイザーがメンバーたちの力量を測れるので、サポートのチカラ加減が想定できるという副次効果もあった。

6.改善による変化や効果

効果として、第一に障害が激減した。さらに、障害事例の理解と情報共有によりテストレビュー時のチェックポイントが充実したこと、経験の浅いS Eの考え方方が成長したことでリスクマネジメント力が向上したことがある。

障害激減について、2013年10月から2014年9月までの障害発生件数が、月に2件から11件だったものが、2014年10月から2015年9月までは、月に0件から3件になった。発生累計での減少率は80%である。

業務遂行環境について、対策前後の一年間を比べてみて、システム変更要求量や仕事量、要員体制は変わらないので、今回の対策が効果をなしたと考えた。ちなみに仕事量の目安の一つとして、出荷判定会の実施件数は対策前の一年間は427件、対策後の一年間は569件であり、対策後の仕事量は勝るとも劣らない。

また、経験の浅いS Eの品質意識が向上したことにより、設計レビュー会、テストレビュー会、出荷判定会などでの、態度・意気込みの真剣さが増し、気づきが多くなったことも、障害の未然防止に役立っている。

7.改善活動の妥当性確認

プラスのスパイラルがまわり始めた。

お客様に本番リリースする前に十分なテスト・検証を行っているにもかかわらず、障害発生件数が思うように減らない状況下、半信半疑でトレーニング参加していたS Eたちも、自らが効果を実感できたのか、職場に活気が戻ってきた。

しかし、ゴールは障害件数をすべての月でゼロ件にすることであるから、現在でも、さらなる要因分析と対策の改善・実施を進めている。

危険予知トレーニング（KYT）は障害予防に役立つ。「こんな要因でこんな障害が起きた」という情報の共有だけでも、有用な知識になることがわかった。また、「こんな条件がそろったときは事故が起きやすい」ということがわかれれば、覚悟と事前準備をしておく、というアクションにつながる。しかし、一番の効果は、「こんなときどんな事故が起きそうか」についての察知力や考え方方が身についてきたことである。

現在、当本部では、この危険予知トレーニング（KYT）の実績を踏まえ、障害の未然防止対策のひとつとして他部門へ横展開中である。

以上

<タイトル>

開発チームと QA がフィードバックし合いながら成長するアジャイル開発の改善事例

<サブタイトル>

アジャイルの特性を生かしたチーム作りと品質の改善

<発表者>

氏名 (ふりがな) : 永田 敦

所属 : ソニー株式会社

<共同執筆者>

氏名 (ふりがな) : 豊田 昌幸

所属 : 株式会社ニッポンダイナミックシステムズ

<要旨・主張したい点>

スクラムマスター (以下 SM) が QA を巻き込み、プロダクトオーナ (以下 PO) と開発チームの間でフィードバックのループを作り、振り返りをしながら改善を進めた。

その結果、QA は、開発中において、ユーザ視点でのシステムテストを行うようになったばかりでなく、PO が作成したユーザーストーリーベースの仕様書のレビューを通じて、ユーザ視点での指摘をし、仕様の理解を深めることができた。さらに、開発チームが、コード実装の前に QA の作成したテスト設計をレビューすることで、開発チームにおいては作りこみの品質が改善され、QAにおいてはテスト設計のモレを補うことができた。

レビューやテストにより QA からフィードバックされる品質情報は、PO や開発チームとの信頼関係を作り上げ、今では、QA が作ったテストケースが、開発チームにおける製品仕様の詳細な振る舞いのリファレンスになっている。

アジャイルの特性のなかで最も重要な、早いフィードバックによる改善の積み重ねに、QA も巻き込むことによって、品質をより早くに組み込み、その中で生まれる活動の質や成果物の質を上げることができた。

<キーワード>

アジャイル、イテレーティブ、品質保証、システムテスト、アジャイルテスト、

<想定する聴衆>

QA、SEPG、ソフトウェア開発者、プロジェクトマネージャ、SM、PO

<活動時期>

2014年6月～

<活動状況>

- 着想の段階 (アイデア・構想の発表)
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他 ()

<発表内容>

1. 背景

2013 年の SPI Japan で“アジャイル開発における品質保証部門によるシステムテストのアプローチ”という題で、アジャイル開発に、早期に QA が入ることによって起こる改善効果について発表しプログラム委員長賞受賞を受賞した。

http://www.jspic.org/event/2013/SPIJapan/session2A/2A3_ID014.pdf

前回は、QA が開発に飛び込み、図 1 のように開発の段階から QA 視点でのシステムテストを行い、その品質のフィードバックを返すことにより、欠陥の停滯時間が少くなり、手戻りの時間が減り、品質を開発の早期から上げていくことに貢献した。

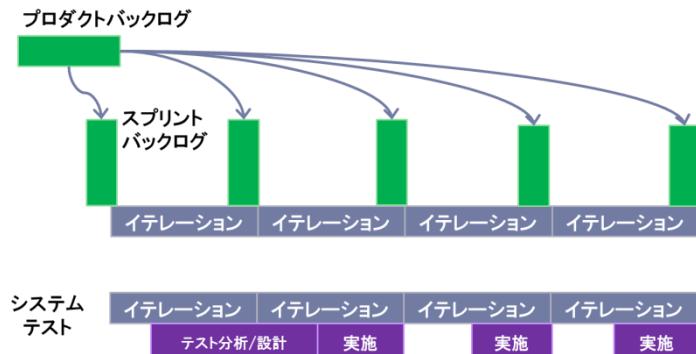


図 1 早期からのシステムテストの実施

また、アジャイルのメトリクスを QA で取り、開発に提供した。これにより、図 2 のようなフィードバックの関係が生まれ、健全な開発活動と品質改善ができ、開発と QA の間での信頼関係が生まれた。

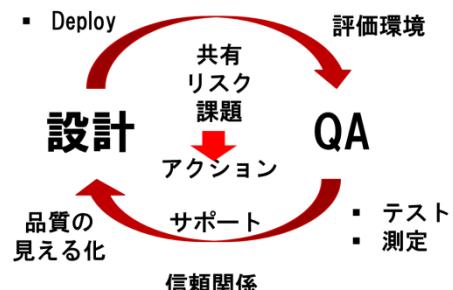


図 1 開発と QA のチームの形成

本発表は、QA がアジャイル開発において効果的な役割を持つことを理解したアジャイル開発チームが、次のプロジェクトでさらに改善を続けた結果、どのように成長したかを紹介し、その改善の意味を考察する。

2. 改善したいこと

改善したいところは以下の 2 点である。

2-1. PO (以下 PO) の仕様の課題

PO が感性に任せて、リッチ（必要以上）な仕様、費用対効果の悪い仕様を作ってしまうことがある。これにより、顧客に対する価値があまり増えない一方で、無駄な要求のために不必要に複雑な設計になり、設計やテストの工数も増え、品質のリスクも増える。

2-2. 開発から QA へのテストベースのフィードバックの確立。

前回のプロジェクトは、QA からの早期のサポートを中心に施策を行った。開発から QA へのフィードバックにおいて、コードのデプロイと評価環境の提供がされたが、QA がシステムテストのテストケースを設計するためのテストベースを、開発がフィードバックとして戻すというところまで確立できていなかった。この、QA がテストに必要な情報を回すプロセスを確立したかった。

3. 改善策を導き出した経緯

まず、2-1 の PO の仕様の課題に対して、

- a. 開発から、開発観点でのフィードバックをもらい、複雑なものを作らないようにしたい
 - b. QA から、顧客観点でのフィードバックをもらい、無駄なを作らないようにしたい
- という要求から、図 3 のような仕様レビューのプロセスを考えた。

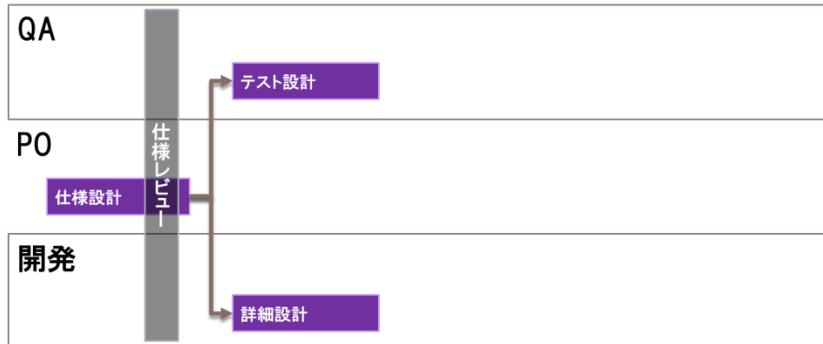


図 3 仕様レビュープロセス

しかし、その矢先に、組織変更があり、別の QA がチームに参入することになった。新しい QA は、以前はウォーターフォール開発での QA 経験者で、図 1、図 2 のような経験、知見を持っていない。

そこで、図 3 のプロセスを行う前に、QA のマインドセットを行い、メンバーにしていく必要が出てきた。そこで、SM がそれを担うことになった。

4. 改善策の内容

以下の 3 つが改善策として逐次勧められた。小さなフィードバックループを作り、QA や開発メンバーを徐々に成長させていくながら進めていくアプローチである。

4-1 スクラムマスター (以下 SM) が QA をメンバーに取り込む：テスト設計と仕様書レビューによるフィードバックループ

QA と PO とのフィードバックループの構築

4-2 テスト設計に対する開発チームのフィードバック

QA と開発チームとのフィードバックループの構築

4-3 ユーザーストーリの Done の定義の工夫

QA によるシステムテストの計画の工夫

5. 改善策の実施

5-1 SM が QA をメンバーに取り込む：テスト設計と仕様書レビューによるフィードバックループ

QA を開発メンバーに取り込むのではなく、SM が QA と一緒に QA をすることにした。言い換えると、最初に行ったのは、QA と SM とのフィードバックループの構築である。

図 3 のプロセスの通り、PO の仕様レビューに QA とともに参加して仕様の理解に徹した。SM はドメイン知識を PO と共有しているので、QA の仕様の理解を助けることができた。

次に、SM が手動でユーザーストーリ単位のテスト設計（マインドマップベース）を実施した。

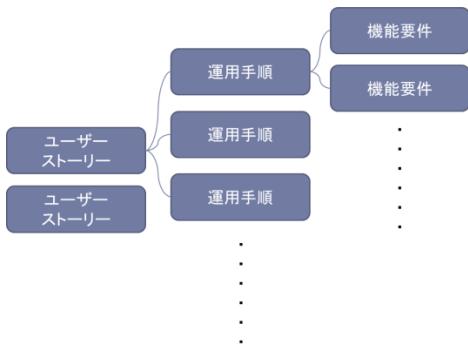


図4 ユーザーストーリ単位のテスト設計のイメージ

しかし、SM は、テストエンジニアではないので、テスト観点からのテスト条件が洗い出せない。そこで、テスト条件は QA にお願いして、設計を完成させた。図 5

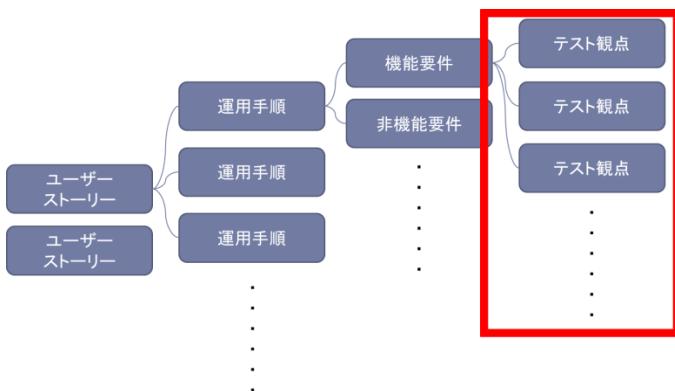


図5 SMとQAで共同でテスト設計をする

これによります、図 6 のような QA と SM の小さなフィードバックループから始めていった。



図 6 OA と SM とのフィードバックループ

今まで、仕様書からのテストケースの作成をしていた QA は、SM と共同でテスト設計し、それに具体的にかかわることにより、徐々にテスト設計スキルを身に着けていくと同時にドメイン知識を吸収していく

このとき、テスト設計で足りないインプットが見つかることがあり、SM は QA の口から PO にフィードバックするよう促した。始めは、一方的に PO の仕様を聞く側に回っていた QA は、これにより、図 7 のような、PO とのフィードバックの関係を作ることができた。

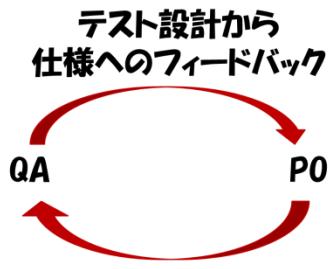


図 7 QA と PO 間のフィードバックループ

これらをまとめると、QA にかかるフィードバックは図 8 のようになる。これを見ると、QA は PO から、仕様に関して、2 つのループから情報をもらい、かつ、各々に質問やコメントを投げている。QA は当初、ドメインの知識不足や、それに伴う遠慮から質問やコメントも少なかった。しかし、フィードバックを回すことで、ドメイン知識を学習し、モノが言える関係が作ってきたので、質問やコメントの量が増え、その質も改善していった。これにより、PO においても、QA からの品質観点からの指摘や、顧客観点からの指摘をうけ、仕様書を改善していった。PO が作る仕様書が QA の成長とともに改善してきたということである。また、PO も的確な質問やコメントを出してくれる QA に信頼を置くようになった。

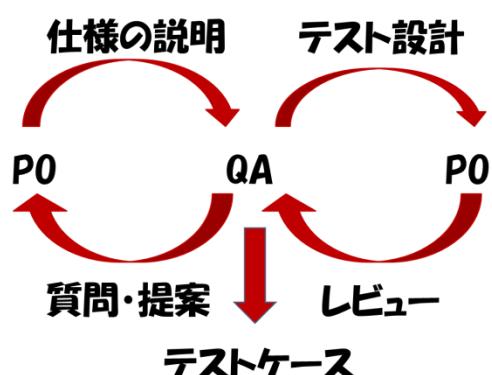


図 8 QA に対するフィードバックループ

このフィードバックループによって、QA がテストしていくうえでの必要十分な情報が得られるようになってきた。つまり、“2. 改善したいこと”で挙げた、“2-2 開発から QA へのテストベースのフィードバックの確立”ができたことになる。これを PO を中心に考えると図 9 のようになる。

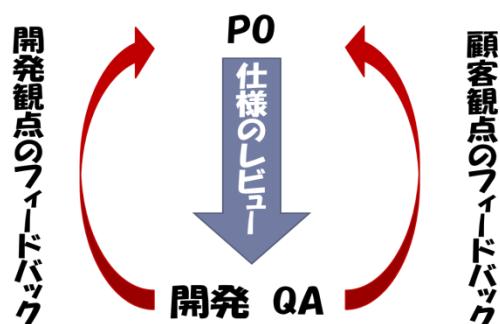


図 9 PO は QA と開発からフィードバックを受ける

図 9 のフィードバックループはだんだん洗練されていき、開発からは、“この仕様よりもこうするともっとシンプルになります” というフィードバックをうけ、QA からは、“顧客は本当にこの機能が必要でしょうか” というコメントまで出るようになった。

これにより、仕様自体が無駄のないものに変わっていた。

ここまで、図 10 のように QA を巻き込んだプロセスにフィードバックが埋め込まれていった。

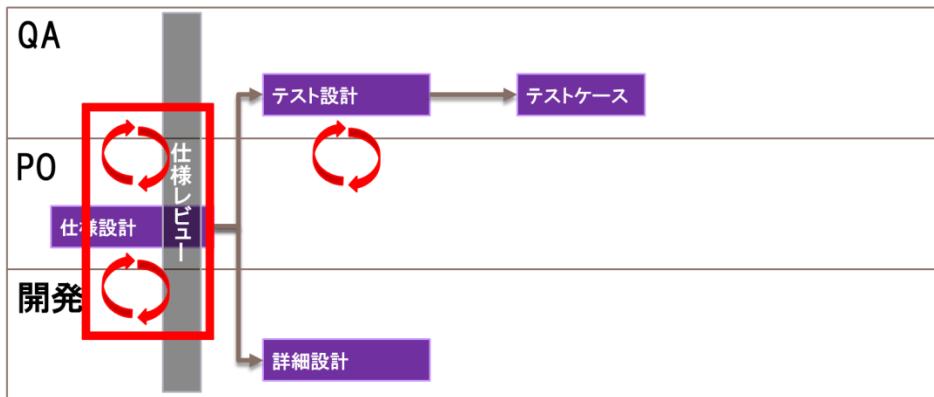


図 10 テスト設計と仕様書レビューでのフィードバックループが生まれたプロセス

5-2 テスト設計に対するフィードバック

その後、もう一つフィードバックを増やしていく。今度は、QA が作ったテスト設計を、開発チームメンバーが設計・実装する前にレビューするようになった。図 11 参照

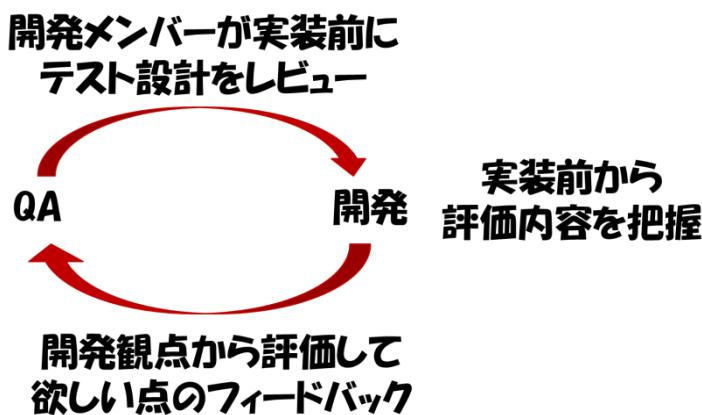


図 11 開発メンバーがテスト設計をレビューする

開発メンバーがテスト設計をレビューすることにより、どの部分をどのようにテストしようとしているかがわかり、自分の気付かなかった異常系のモレなどを実装に盛り込むなどの品質の補強ができる。また、逆に、開発メンバーが、このようなところが気になるのでテストしてほしいというようなテストの追加要求を伝え、テストのモレを防ぎ、テスト効率を強化することが出来る。

今まで述べたことを開発プロセス図であらわすと図 12 のようになる。赤いループで、QA, 開発チームでレビューによりフィードバックがかかっていることが分かる。

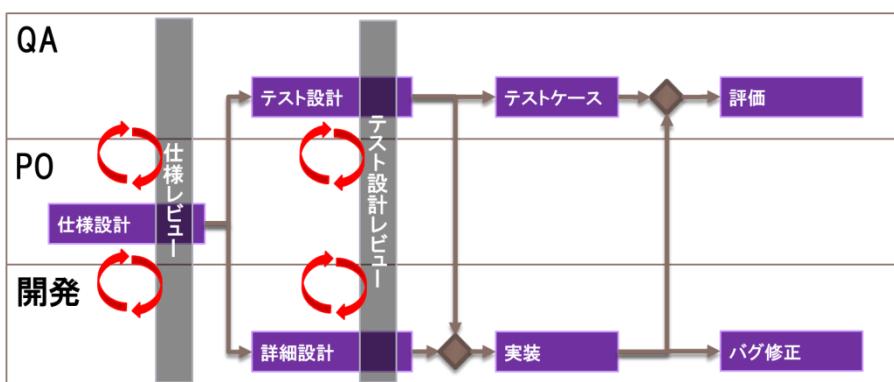


図 12 開発プロセスにおけるフィードバック

5-3 QAによるシステムテストの計画の工夫

当初、ユーザーストーリーの Done (完了) の定義を、QA テストが完了し、かつ Bug が 0 であることとしていた。図 13 のように、開発ごとに、機能テストばかりでなく、必要に応じて負荷テスト、長期安定性、性能テスト、ワークフローテストなど、時間のかかるテストまで行い、合格することが Done の定義としていた。

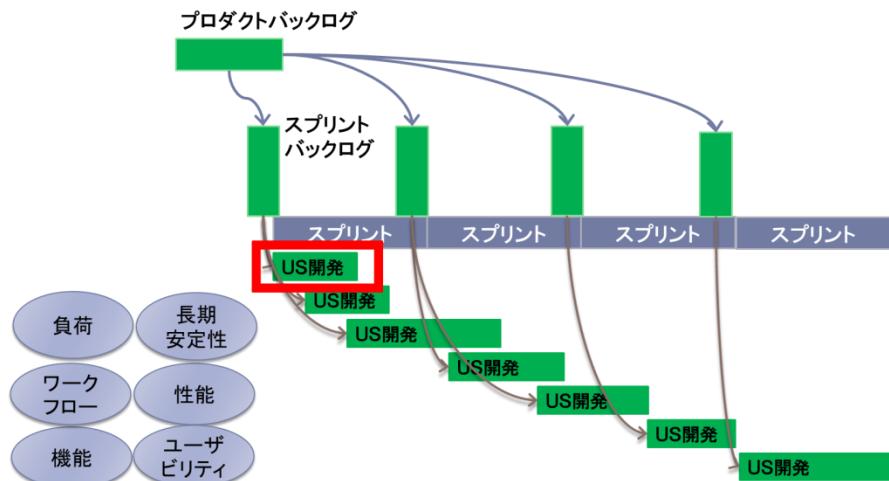


図 13 当初の QA のテストのイメージ

しかし、負荷テストや長期安定性、パフォーマンステストなどは時間がかかることがあるため、それが完了するのを待つと、なかなかストーリーが Done しないことになり、進捗が見えなくなってしまった。また、QA にも、早く終わらせるためのプレッシャーがかかり、疲弊してしまった。そこで、Done の定義を変えて、図 14 のようにテストスプリントを設けて、進捗の健全化と確実なテストの実施を両立させた。

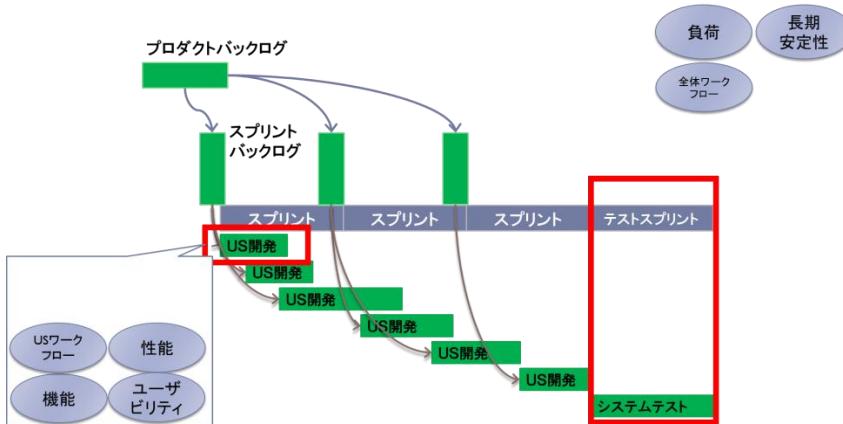


図 14 Done の定義の変更とテストスプリントの追加

6.改善による変化や効果

QA のテストケースが、製品の詳細の振る舞いの仕様書となり、開発チームはそれを開発のレファレンスとするようになった。

つまり、図 15 にしめすようにテストケースが神様として扱われ、そのあとの PO や顧客などへのデモにおいてもリファレンスとなった。

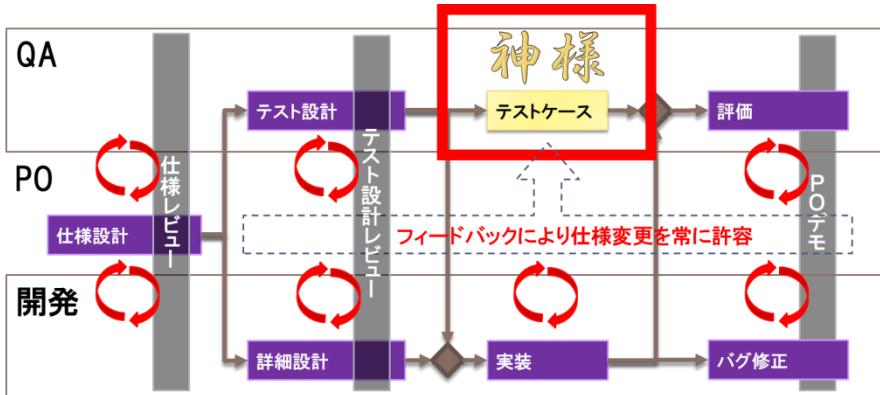


図 15 現在の開発プロセス

これにより、QA は自分の作った成果物に対しての責任を自覚し、PO や開発チームとの連携を高め、ドメイン知識の充実、顧客視点やソフトエンジニアリングの研鑽、テスト技術の向上のモチベーションが高まった。QA は PO や開発チームと同等に互いにリスクペクトし、コミュニケーションがより密になった。

ポイントは、図 15 に示されるプロセスは、一晩でできたものではなく、2 年の月日を重ねて、それぞれのフィードバックループを充実させ、それぞれの成果物の品質を上げていった結果できあがったものである。当初の PO が作った仕様は、そこに書くべき異常系の機能記述や非機能特性の定義なども漏れていた。その仕様書は QA からのフィードバックにより、改善していったのである。

また、QA の方は、仕様書レビュー、テスト設計レビューにおけるフィードバックの中で、ドメイン知識の詳細情報を学習し、テストに必要な情報を十分得られるまでに成長することができた。それにより、レビューのフィードバックや、テストのフィードバックによる品質面でのサポートを開発チームに提供することにより、開発チーム、PO との信頼関係を持つことが出来た。これにより、図 16 に示すようなより理想的なアジャイルの姿になった。

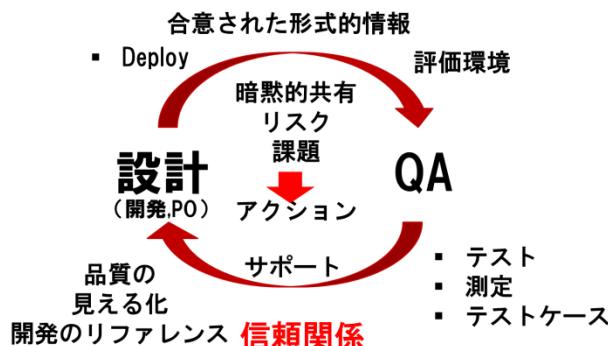


図 16 より理想的なアジャイルの姿

7. 改善活動の妥当性確認

図 15 に示すプロセスになったことで、さらには、開発や QA からのフィードバックから、無駄のない、価値を明確に説明できる仕様になり、初めから無駄な作業が起こらない品質に変わっていた。これにより、2-1 で上げた、PO (以下 PO) の仕様の課題を解決することができた。

2-2. 開発から QA へのテストベースのフィードバックの確立については、図 15 のプロセスを作る過程で、図 8 のフィードバックが作られ、QA の学習レベルが上がるにつれてフィードバックの質が上がることにより、QA が必要なテストベース情報を得られることが確認された。開発チームが、その情報によるテスト設計から作られたテストケースを、製品の詳細な振る舞いの“神様”とみなしたのは、QA が仕様書レビューを経て得てきたテストベースの妥当性、そして、開発チームのレビューをうけてきたテスト設計の妥当性を理解しているからに他ならない。

このように、フィードバックループを“設計”し、それを改善、成熟させることにより、フィードバックループに携わる人のレベルをあげ、そこから生まれる成果物の品質と効率を改善することが出来ることを示した。これこそがアジャイルの醍醐味であり、アジャイルの本質的なスピリットである。そう考えると、すべてのアジャイルプラクティスは、改善のための手段であることが分かる。今回の事例により、

それが実証され、より意味を理解することができた。また、この活動をドライブしていたのは SM で、その役割は、アジャイル開発の改善において非常に大きいことが分かった。

図 1 5 のプロセスは、単なるスナップショット、通過点に過ぎず、チームメンバーは、振り返りを繰り返して、さらに変革していくことを確信している。これは、楽ではないが、楽しく生き生きとチームが活動していることを意味し、アジャイル開発として成功しているといえる。

A. 参考文献

Competitive Engineering, Tom Gilb, 2005

知的生産企業、野中郁次郎、竹内弘高,1996

Changing Minds, David Straker, 2008

斥候としてのアジャイルプロセス活用の提案、細谷泰夫、2012、

http://www.jaspic.org/event/2012/SPIJapan/session2B/2B2_ID008.pdf

2B1 「全員参加型画面レビューのススメ」 長谷川真裕(インテック)

<タイトル>

全員参加型画面レビューのススメ

<サブタイトル>

デザイナーがいないチームで UX を向上するための数年に渡る取り組み

<発表者>

氏名 (ふりがな) : 長谷川真裕 (はせがわまゆ)

所属 : 株式会社インテック

<共同執筆者>

氏名 (ふりがな) :

所属 :

<要旨・主張したい点>

デザイナーがいない、派生開発において画面開発を成功させるには、全員参加型で画面レビューすることが有効である。

画面レビューは新機能・変更機能に対するレビューと他機能への影響がないかのレビューの 2 種類がある。

確認ポイントや操作方法などを明記したり、あらかじめすべての機能を網羅するような設定情報を準備したり、ステージング環境で標準ブラウザを用いることでいつ、どこで、だれがレビューしても抜け、漏れがないようにできる。

画面開発者だけではなくプロダクトマネージャーやエンジニア含め全員が担当することで UX の向上にも寄与できる。

<キーワード>

- * UX
- * UI
- * 画面レビュー
- * ステージング環境
- * CI
- * 全員参加型
- * CSS フレームワーク

<想定する聴衆>

UI 開発をする若手のソフトウェアエンジニア

デザイナーがいないから優れたデザインができないと悩んでいるプロダクトマネージャー

画面開発者がいるチームのすべての関係者 (API 開発エンジニアや導入メンバーなど)

<活動時期>

2008 年～継続中

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

主体：

自社製品開発において業務システムの理者用画面、エンドユーザー用画面の開発をおこなっているアプリケーションエンジニア

開発未経験の状態で 2008 年から開発に携わり、Java を用いたデスクトップアプリケーションの開発を 4 年程度、現在は RubyOnRails を用いた Web アプリケーション開発を担当している。

現在、チームは全員で 5 人。プロダクトマネージャー、バックエンドアプリのエンジニアが各 1 名、フロントエンドアプリのエンジニアが 3 名である。

動機：

デザイナーがいない中、画面開発をおこなう上で使いやすさを向上するためには、レビューを適切に行うことが重要と考え、サブリーダーによるトップダウン型式のレビュー、週一程度の全体レビュー会、モンキーテスト大会などをおこなってきた。

また、OSS の CSS フレームワーク（本プロジェクトでは Twitter Bootstrap）を利用してすることでソースコードの均一化と開発スピードの向上を図った。

2.改善したいこと

チームの成熟とともにこれまでのレビュー方法が合わなくなり辞めてしまったり、単体テストが通ることだけでレビューを終わらせてしまったりすることもあった。また、個々のエンジニアが各々の好みで UI を作りこんでしまい画面によってばらつきが出たり、既存の機能に悪影響があることに気がつかなかったり、UX 上の改善点を共有できなくなってきたり問題も出てきた。

派生開発において追加、変更される機能単体のレビューをおこなうだけではなく、当初のデザインでは想定していなかったデザインを検討、それによる製品全体の UX を低下させないようにする必要があった。

なお、UX とは「ユーザーが製品・サービスを通じて得られる体験」のことである。本稿ではエンドユーザーが私たちのソフトウェアを使用してすでにある業務を円滑におこなうことができること、という観点で使用している。

3.改善策を導き出した経緯

場当たり的、形式的だったレビューのやりかたを見直し、UI、しいては UX 上の課題を捕捉できるような画面レビューの方法を考えた。解決すべき課題は以下の 3 点である。

- ・新規機能や変更機能以外に既存の機能に影響がないか確認できること
- ・一部のメンバーにレビュー負荷がかかりすぎないこと
- ・既存の開発プロセスと親和性の高いこと

また、派生開発では追加する機能、変更する機能だけにフォーカスしてしまいがちだが、全体的なユーザビリティとの整合性も維持しなければならない。

4.改善策の内容

プロジェクト自体はアジャイル開発でおこなっていたので、アジャイル開発（イテレーション開発）にあった全員参加型の画面レビューをおこなった。

私たちのチームはタスクを Trac のチケットを用いて管理している。タスクが終了すると画面や機能などのレビューを他のメンバーに依

頼する。このチケットを用いて画面レビューを行うことを考えた。レビューのための環境は、CI（継続的インテグレーション）により毎日、ビルド、デプロイしている環境(ステージング環境)を活用した。また、イテレーションの区切りやシステムテスト期間を利用して全体的な画面レビューをおこない次バージョンへの改善すべき課題を棚卸しした。すでにあるプロセスを活用することで画面レビュー 자체も自然におこなわれるようと考えたのだ。

レビュー時はできるだけ実際に動くものを見て判ることを重要視した。言葉だけでは皆がバラバラのものを頭に思い浮かべることがあり、指を差せる何かが必要なのである。

5.改善策の実現方法

日々おこなうものとシステムテスト時におこなうものがある。まず、レビュー依頼時、担当者はチケットに確認してほしいポイントや操作手順、データなどレビューに必要な手順、条件、情報を提示する。レビュー者はこれらの情報をもとにレビューをおこなっていく。次に、全体レビューとしてシステムテスト期間には全員が集まり各画面を確認する。抜け漏れがないようにすべての機能を網羅するための設定をあらかじめ準備したり、想定シナリオを列挙したりしておく。

ユーザーが意図しない使い方をしても問題がないか確認するためのモンキーテストも継続しておこなった。

6.改善による変化や効果

以下の3つの効果があった。

- ・レビューノウハウを共有しながらレビューできる
- ・相談や質問など成果物を指差しながらコミュニケーションできる
- ・多様な視点でレビューできる

レビュー内容は蓄積されているので時間が経過した後でもチケットを見返すことで当時の状況を思い出すことができる。レビュー中も実物を動作させながらコミュニケーションできるため、担当者とレビュー者との間で認識のズレが発生しなくなった。担当者やレビュー者の思い込み、新たな顧客ニーズへの気づきにつながることもあった。

さらに、バグ発覚時の原因の切り分け、再現方法の調査や、マニュアル、チュートリアル作成時にも利用できた。

7.改善活動の妥当性確認

全員参加型レビューをおこなうことにより画面開発における品質の維持は一定の効果があった。

様々な役割のメンバーが参加することでレビューする視点が広がり、ビジュアル面だけではなく技術面や UX 面など多くの気づきが得られた。

ただし、ソースコードの品質上は課題も出てきた。レビューでは成果物にばかり注目してしまい新しい UI デザインが増えたことにより個別のスタイルがどんどん増え、また CSS フレームワークを無視した使い方や既存のスタイルが悪影響を及ぼすのか important! が頻出するようになったのだ。

そのためにデザイナーでなくてもデザインを担当するメンバーを任命することが好ましいと考えている。このメンバーはファシリテーターとしての役割を担い、ルールに従ったスタイルが適用されているか定期的にチェックする。また、CSS フレームワークを全メンバーが理解するために定期的に勉強会を開催することも検討したい。CSS フレームワークを正しく使うことでその CSS が期待している動きなどが理解しやすくなり、テストがしにくくデザインの部分のリファクタリングもしやすくなると期待している。

今後は使いやすさと並行して CSS などのソースコードの品質を保つための開発プロセスなども検討したい。

A. 参考文献

デザイン・ウェブナビゲーション —最適なユーザー・エクスペリエンスの設計 (2009/5/25)

(著) James Kalbach (監訳) 長谷川 敦士、浅野 紀予 (翻訳) 児島 修

インターフェースデザインの心理学 —ウェブやアプリに新たな視点をもたらす 100 の指針 (2012/7/14)

(著) Susan Weinschenk (翻訳) 武舎 広幸、武舎 るみ、阿部 和也

みんなではじめるデザイン批評—目的達成のためのコラボレーション&コミュニケーション改善ガイド (2016/5/24)

(著) アーロン・イリザリー、アダム・コナー (翻訳) 安藤貴子

2B2 「作りすぎない基幹システム刷新プロジェクト」伊藤茂泰(富士通コワーコ)

<タイトル>

作りすぎない基幹システム刷新プロジェクト

<サブタイトル>

～現場利用者を巻き込んだ要件確定～

<発表者>

氏名 (ふりがな) : 伊藤 茂泰(いとう しげやす)

所属 : 富士通コワーコ株式会社

<共同執筆者>

氏名 (ふりがな) :

所属 :

<要旨・主張したい点>

今回発表する基幹システム刷新プロジェクトは、メインフレームからクラウドへの刷新である。全く異なる基盤の中で、現場が日々利用している膨大な開発資産から機能を最小限に抑えクラウドで稼働させるためには実際に業務が継続できるかという視点で、開発、運用、現場がワンチームとなって検証できるプロセスが不可欠である。本発表では実際に現場を巻き込みながら業務フローを検証していった効果と、その気づきを紹介する。

<キーワード>

ERP パッケージ、アジャイル開発、クラウド、大規模

<想定する聴衆>

パッケージ導入・クラウド導入を検討されている方

これからアジャイル開発に挑戦しようとする方

<活動時期>

2011年1月～2012年3月 Fit & Gap～

2012年4月～2013年5月(開発～稼働)

<活動状況>

- 着想の段階 (アイデア・構想の発表)
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他 ()

<発表内容>

1.背景

弊社基幹システムはハードウェアの老朽化、維持コストの削減、新規ビジネスへの柔軟な対応といった様々な面から次期システムへの置き換えの必要に迫られていた。当基幹システムは規模が大きく、外部システムとの連携も多岐に渡っており過去に一度パッケージ適用で失敗(断念)している。もう失敗が許されないというプレッシャーの中、弊社は従来のメインフレームからクラウド(Salesforce+GLOVIA OM(ERPパッケージ))への基幹システム刷新プロジェクトを発足した。

今回のプロジェクトに於いては外部インターフェースを変更しない事を方針とし、フロント部分に関してまず全体の業務の流れを整備した上で作る範囲を明確にし、その後アジャイル開発のスクラムで開発を進めた。また、企業内SNSを活用した課題の見える化と検討、運用、補助ツールの内製化等を実施しながらプロジェクトを推進した。

2.改善したいこと

今回のプロジェクトは単なるリプレースではなく、刷新を目的とした。

そこで膨大な現行システムの機能から新基幹システム上で実現するものを必要最小限に抑え、新基幹システムで実現する範囲、また操作イメージを早い段階で現場と共有しながら進めていきたいと考えていた。

3.改善策を導き出した経緯

前回のパッケージ導入での失敗の要因としてはパッケージに業務を合わせるという方針から、現場の利用者がほとんど関与しておらず、実際の画面を見てから利用者からの改善要望が殺到し、対応しきれなかつた事が一つに挙げられる。

日々の業務で利用する基幹システムの環境が変わるという事はそれだけで現場にとっては非常にストレスや不安が大きい。また、以前の基幹システムの画面は全てVisual Basic5.0で作成されていたためWeb画面に移行する事により画面の操作性が大きく変わるというインパクトも大きい。

上記はドキュメントベースでは伝わりにくい事も多く、実際にどのように変わるのがというイメージを早い段階で現場に伝える事でゴールへ向けて全員のベクトルを合わせ、本来の優先順位を上げるべき課題に集中できる環境を作る必要があった。

4.改善策の内容

FIT&GAP工程に多くの期間(1年)をかけて、新システムにおける業務の流れを検証した。

現場へのヒアリングを元に作成したAs-Is業務フローからこれをSalesforceで実現した場合のTo-Be業務フローを作成、これをベースに委託先の開発チーム、弊社側(システム部門、各現場代表)で一箇所に集まり、動く画面を見ながらTo-Be業務フロー及び画面の検証、修正を繰り返し実施した。

画面の検証においては、(工数小)Salesforceでの標準画面 → 標準でカスタマイズできる範囲 → サードベンダーの提供する画面生成ツール → 個別開発(工数大)といった段階的なアプローチを踏む事で実際の要望の難易度によっての実装レベルについて全体で合意した。

レビューにおいては、議論が広がらないように「画面はSalesforceの標準画面をベースとする」「個別業務は極力無くしていく」といった方針を立て、トップダウンのメッセージとして先に伝えた。

5.改善策の実現方法

最初は一台のプロジェクターに新しい(Salesforceでの)画面を投影しながら現場とのレビューを実施していたが現場は以前の基幹システムしか利用していないため、イメージが伝わりにくかった。そこで次回からはプロジェクターを2台準備し、新旧2つの画面を同時に投影しながら議論を実施したところ

現場からのフィードバックが出やすくなつた。

その反面、「画面は Salesforce の標準画面をベースとする」という方針はあったものの、ここだけは譲れない(標準画面では対応不可)というフィードバックも多々あり調整に苦労したが、利用頻度や実現コストの観点で交渉の上、極力標準画面での実装を促した。

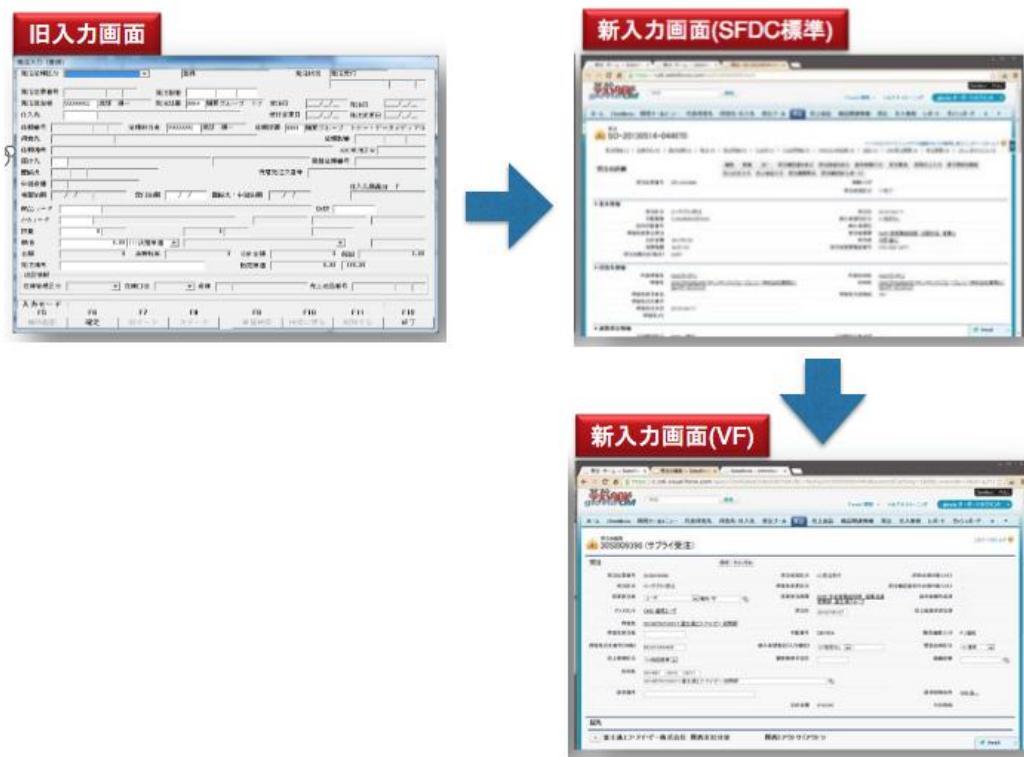


図 6 旧画面と段階的アプローチ

レビューの中でビジネスの変化に合わせて現場の利用者が現行システムを工夫して使っているため As-Is が複雑になっているケースや、属人化している業務にも気づくことができ、その部分については根本から見直すことでしっかり To-Be に反映させていった。逆にマスタの変更画面や帳票類においては多くのもの Salesforce 標準で持つレポート機能でカバーできることで合意ができ、開発の抑制につながつた。

レビュー会に於いては、現場の実務担当者は日々の業務に追われており、なかなか同じ時間に集まる事が困難で必要な時に必要なメンバーが集まれない事が多かったため、レビュー会の中で時間を細かく区切り現場のちょっとした空き時間でできるように工夫した。

6.改善による変化や効果

4,5 の改善策により、To-Be 業務フローを軸に実際に基幹業務を継続できるか?という観点での議論に集中する事が出来た。また現場とは単なる旧画面の操作の解説だけではなく、その背景についてもしっかり議論が出来たため開発側も実際の業務への理解を深めることができた。

開発を最小限に抑える事が出来たため、1/3 の開発コスト削減、1/2 の期間で構築し、当初の予定通り 2013 年 5 月のゴールデンウィークに切り替え、Salesforce 上で基幹システムを稼働させる事が出来た。

心配していた画面については実際に稼働させてみたところ現場がすぐに操作に慣れ、特に改善の要望もなく、標準画面のまま運用ができている事から作りすぎを抑制できた。

2013 年 5 月稼働後、基幹システムと連携した新しい価値を段階的にリリースできている。

7.改善活動の妥当性確認

今回の開発に於いては、基幹という大規模案件において Salesforce の標準での開発スピードの速さの特性を活かして FIT&GAP 工程の段階から現場利用者を交えて、実際に動く画面で検証できた事によって、開発するもの、開発しないもの(標準機能で十分対応できるもの)の明確化、現場利用者との合意につなげる事が出来た事が予定どおり基幹システムを稼働させる事に繋がった。

また、本番稼働時まで常に実際に動く画面を公開していたため、本番切替後も現場は混乱することなく新しい基幹システムで業務を再開できたことも良かったと考える。

優先順位付けのための基準(方針)については全ての関係者で合意のため、トップダウンで予め伝えておく事が望ましい。これにより優先順位付けの理由が明確になり納得性が高まった。

A.参考文献

<タイトル>

TOCに基づく日立グループ 1,000 人に広がる業務改善 6ヶ月プログラム

<サブタイトル>

はじめやすく成果を生む改善プロセス

<発表者>

氏名 (ふりがな) : 八木 将計 (やぎ まさかず)

所属 : 株式会社日立製作所

<共同執筆者>

氏名 (ふりがな) : 西原 隆 (にしはら たかし)

所属 : ゴール・システム・コンサルティング株式会社

氏名 (ふりがな) : 真道 久英 (しんどう ひさとか)

所属 : ゴール・システム・コンサルティング株式会社

氏名 (ふりがな) : 村上 悟 (むらかみ さとる)

所属 : ゴール・システム・コンサルティング株式会社

氏名 (ふりがな) : 渡辺 熏 (わたなべ わおる)

所属 : 株式会社日立製作所

<要旨・主張したい点>

本報告では、残業削減や生産性向上といった業務改善を効果的かつ効率的に行うため、TOC (Theory of Constraints)思考プロセスに基づく標準フォーマット「チームマネジメント A3」による、タスクボードと朝会を用いた業務改善プログラムを提案する。本手法では、チーム内の推進者が改善支援者と面談し、チームマネジメント A3 に基づいて自チームの改善を検討する。タスクボードと朝会により、日々 15 分程度の時間がかかるが、改善現場の負荷が高くないため、はじめやすく、成果を生みやすいという特徴がある。結果、助言を得やすくなったりなどの定性的効果だけではなく、残業時間 30% 減や納期遵守率 27% 増などの効果も得た。また、標準フォーマットを用いることで改善支援者の負荷も小さくなり、現在、設計開発を中心として、のべ 1,000 名の規模の改善活動に展開できている。

<キーワード>

業務改善、タスクボード、朝会、チームマネジメント、TOC(Theory of Constraints)、TOC 思考プロセス、抵抗の 6 階層、合意の 6 ステップ、KPT

<想定する聴衆>

SEPG などの改善支援者

<活動時期>

継続中

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

ソフトウェアプロセス改善に限らず、一般的に業務改善活動はうまくいかないことがある。業務改善が必要ということは、その現場は多忙であることが多い。それを解消するために、さまざまな改善活動が行われるが、それらはスキルが必要など、簡単には導入できないことも多いため、さらに忙しくなるという悪循環に陥る。

また、人の変化に対する抵抗には6階層あるといわれており、その中の第1番目の抵抗は、「問題に合意しない」であるといわれている[1][2][3]。特に、トップダウンで改善方法を取り入れようすると、実行する現場の担当者が問題に合意していないことも多く、改善活動が正しく実行されなかったり、形骸化したり、効果が出ないことにつながってしまう。一方、現場が主導するボトムアップでの改善は、現場の抵抗は低くなるが、現場のスキルに依存するため、改善範囲が小さくなってしまったり、的外れな改善策を選択したり、場合によっては改善効果が出ないこともある。

そこで、このような現場の改善活動を正しく導くため、SEPGやPMOを設置したり、外部のコンサルタントと提携したり、改善の支援がある。しかし、このような形での業務改善を支援する方法では、個別事情に合わせた支援が必要になるため、社内へ広く展開しづらいという問題もある。

そこで、本報告では、効率的かつ効果的な改善活動を行うため、作業タスクの見える化ツールであるタスクボードと、TOC(Theory of Constraints)の思考プロセスに基づく標準フォーマット「チームマネジメントA3」による業務改善プログラムを提案する。

2.改善したいこと

2.1 忙しさを増幅する問題構造

ソフトウェアプロセス改善に限らず、業務改善は、残業が多い、作業効率が悪い、ワークライフバランスが悪いなどといった問題があるため、多くの場合、業務改善を実施する現場は「忙しい」ということがいえる。

このような忙しい状況を整理すると図7に示すような問題構造を持つ。トラブル対応や会議などで忙しい状況であるため、仕様書や手順書などのドキュメント作成が不十分になることがある。そうすると、必要なノウハウが共有できず、特定個人のメンバーに仕事が集中することになる。結果として、過負荷状態になるため、納期を遅延し、最悪の場合、顧客からのクレームにつながる。また、ドキュメントが不十分であると、製品・サービスの品質を保てなくなるため、障害が増え、クレームにつながり、最終的にはトラブル対応に戻ってくるため、悪循環になる。さらに、そのように忙しい状況がつづくと、残業や休日出勤が増え、現場のモチベーションが低下する。また、教育にも時間をさくことができなくなるため、人材が育たなくなり、モチベーション低下や特定個人への仕事の集中につながる。

このような問題構造の中、業務改善のため、問題の解決につながらないさまざまな施策が打たれることが多い。例えば、品質低下に対して、品質向上委員会を設置したり、ノウハウが共有されない状況に対して、ナレッジデータベースを構築したり、教育できていない状況に対して、教育委員会を設置したりする。図7に示すようにそれぞれの事象が関係している中で、このようなさまざまな取り組みを行うことは効果が小さいだけではなく、「忙しくて時間がないのにも関わらず、さらにさまざまな活動に時間を取られてしまうため、忙しさを増幅することにつながる。

つまり、業務改善を進める上では、まず「時間を生み出す」ことが必要といえる。

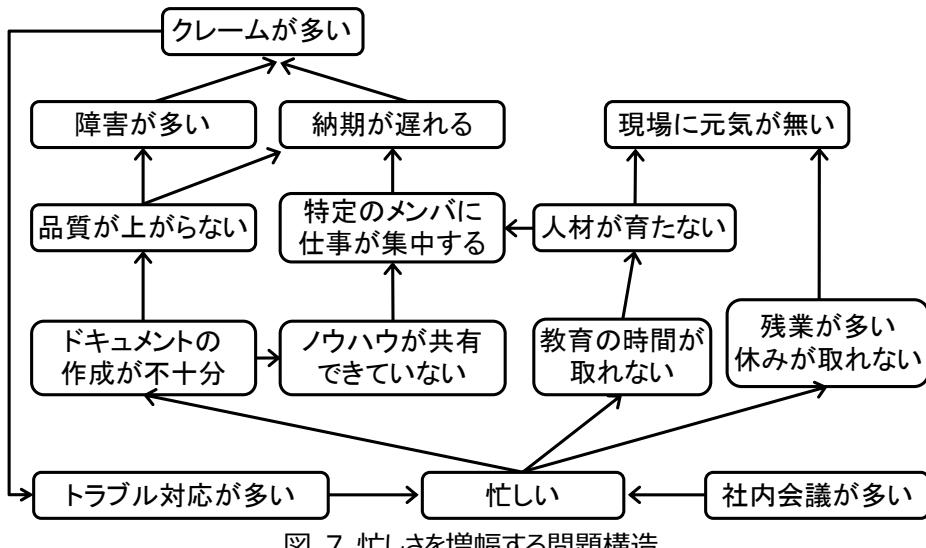


図 7 忙しさを増幅する問題構造

2.2 アプローチに起因する問題

人が業務改善のような「変化」に抵抗するときの心理は、表 2 に示す 6 段階の階層構造になるといわれている[1][2][3]。この抵抗の 6 階層の詳細は以下のとおりである。

【第 1 階層】問題の存在を認めない

「問題」の捉え方の違いに起因する抵抗。そもそも「問題」とは思っていないものを他者から指摘されても素直に認めることはできない。「問題ではない」という意識は強く根深い場合が多く、抵抗の最初の階層として位置付けられる。関係者間で別々の問題認識をしている場合、解決そのものも実行困難である。

【第 2 階層】解決策の方向性に合意しない

問題の存在に合意が得た後に発生する抵抗は、解決の方向についての意見の相違に起因するものである。特に複数の部門が関係する問題では各人の所属部門の利益を優先し、解決策の方向性に抵抗を示すことが多々発生する。

【第 3 階層】解決策が問題を解決するとは思わない

提案されている解決策が本当に問題を解決できるかという疑問に起因する抵抗である。「ウチは特別だから、その方法では解決できない」といった類いの抵抗もここに含まれる。

【第 4 階層】解決策を実行すると副作用が生じる

解決策の実行によって発生すると予想できる副作用も抵抗の大きな理由となる。例えば、「実行すると効率が下ってしまう」「実行すると工数が増えてしまう」といったものである。

【第 5 階層】解決策の実行を妨げる障害がある

副作用がないことがわかると、次に実行を阻害する障害の懸念が発生する。「技術力がない」「社内ルールがあつてできない」といったものである。

【第 6 階層】未知のことへの恐怖感がある

人が持つ本能的な「変化」への恐怖心に起因する抵抗である。例えば、「実行すると何が起こるかわからないからやらない」といったものである。

表 2 変化に対する抵抗の 6 階層と合意プロセス

階層	抵抗の 6 階層	合意の 6 ステップ [°]
1	問題の存在を認めない	問題に合意する
2	解決策の方向性に合意しない	解決策の方向性に合意する
3	解決策で問題を解決できると思わない	解決策で問題が解決できることに合意する
4	解決策を実行すると副作用が生じる	解決策の実行で重大な副作用がないことに合意する
5	解決策の実行を妨げる障害がある	解決策の実行を妨げる障害を克服する方法に合意する
6	未知のことへの恐怖感がある	未知のことへの恐怖感の克服する

つまり、業務改善がうまくいかない要因の一つは、抵抗の 6 階層のいずれかに引っかかってしまっていることである。

改善施策をトップダウンで決定し、現場に実行するトップダウン的なアプローチを取ると、改善の結果を重視し、深く根本原因を追求せずに、手をつけることが多くなりやすい。そうすると、メンバーが問題に合意できず、対立が生じ、命令・指示が増え、やらされ感が出て、言われたことだけをやる状態となりやすい。すると、想定どおりの効果が出ず、犯人探し始まり、現場がギスギスだ。結果、改善する意欲が薄れるため、同じことを繰り返す悪循環に陥る。

一方、抵抗の 6 階層は、第 1 階層から順番に取り除くと受け入れられやすくなるといわれている(表 2 の合意の 6 ステップ)。自律改善などのボトムアップ的なアプローチが第 1 段階から順番に合意していく方法となるが、現場の抵抗は低くなるものの、実際の改善は現場のスキルに依存するため、改善範囲が小さくなってしまったり、的外れな改善策を選択したり、場合によっては改善効果が出ないことも発生する。

2.3 業務改善支援上の問題

前節に示すとおり、業務改善では、現場の抵抗を考慮しながら進めることが重要である。そのため、日立では、抵抗の 6 階層に沿った問題解決を行う TOC(Theory of Constraints)[4]の思考プロセスを業務改善の支援として採用している。具体的には、TOC コンサルティング会社と提携し、ボトムアップ的アプローチを基盤とした業務改善の支援を行っている。

業務改善の対象である現場では、それぞれ個別の事情を抱えている。TOC 思考プロセスに基づく業務改善では、それら個別事情に対応しながら進める形になるため、改善支援者であるコンサルタントの負荷が高くなる。この改善支援が行える人的リソースには限りがあるため、結果として、日立グループ全体における TOC に基づく業務改善の取り組みの展開を制限されている問題があった。

3. 改善策を導き出した経緯

ソフトウェア開発は、単純なルーチンワークではないため、どこかでトラブルなどの「問題」が発生し、その対応が必要になる。一般的に業務上で発生した問題は図 8 に示すようなライフサイクルで解決される。つまり、問題が発生した後、チームの誰かがそのことに気づき、チームもしくは上司に報告・相談しなければならないと感じ、共有される。その後、対策の検討が決定し、対策案が決まり、実行されて、問題が解決する。

この問題解決のライフサイクルにおいて、問題解決に有効な「対策期間」は、問題が共有された後から解決策が実行されるまでとなるが、発見から報告までの期間は、問題が発生しているのにも関わらず、問題解決に対して有効な手立てが取られていない「潜伏期間」となっている。特に、仕事への責任感が強い人ほど、問題が自分の手に負えない状況になるまで問題を抱える傾向があるため、潜伏期間が長くなる。

これらのうち、対策期間はそもそも難しいことも多く、短縮することが難しい。一方、潜伏期間は、チーム内のコミュニケーションの習慣を変えることで短縮することが可能である。

そこで、次章以降に示す提案手法では、この潜伏期間を短縮して、改善のための時間を作り出すために、次章に示すタスク

ボードと朝会[5][6]を用いる。そのように改善の時間を作り出した上で、抵抗の6階層(合意の6ステップ)に対応したTOC思考プロセス[1][2][3]をタスクボード＆朝会向けに標準化することで効率的かつ効果的な業務改善プロセスを導いた。

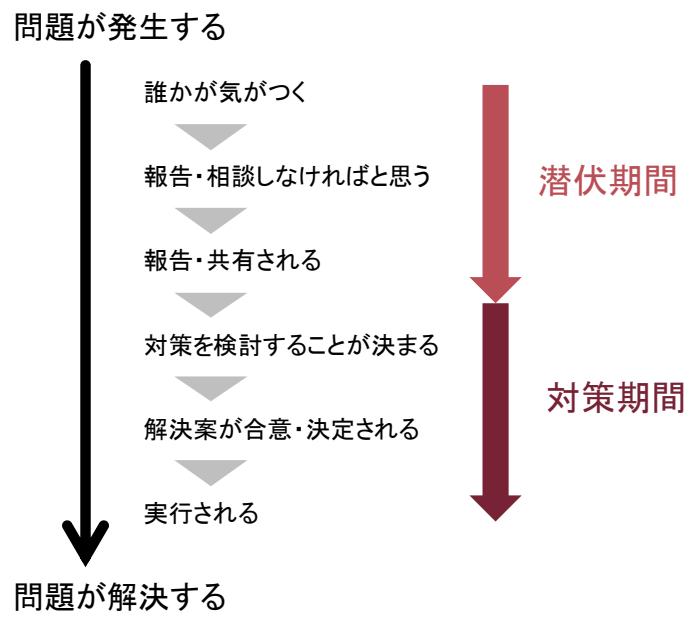


図 8 問題解決までのステップと潜伏期間

4.改善策の内容

4.1 アプローチの方向性

前章に示すとおり、業務改善を進める上では、まずタスクボードと朝会(図 9)で時間を作り出し、その上で、抵抗の6階層に基づいて適切な改善へと導くため、TOC思考プロセスに基づく「チームマネジメント A3 (図 10)」による業務改善プログラムを提案する。それぞれの目的・内容は、以下である。

- タスクボードと朝会
 - ✓ 目的
 - ✧ 改善のための時間を作り出す
 - ✧ チームマネジメント A3 による改善ポイントを得る
 - ✓ 内容
 - ✧ タスクボードとは、その週の予定タスクを貼る「ToDo」欄、本日実施しているタスクを貼る「Doing」欄、完了したタスクを貼る「Done」欄を最小限とした、ボードである(図 9)。なお、欄はチームごとにカスタマイズする。
 - ✧ チームメンバー各自のタスクをタスクボードに貼り、スタンドアップミーティング(通称「朝会」、必ずしも朝である必要はない)で共有することで問題解決を迅速にする。運用の詳細は 0 節参照。
 - ✧ タスクボードの運用が定着化すると、現状のチームの業務状況がわかるようになることで、チームにおける問題点、つまり、改善ポイントが明確になる。改善ポイントの例は 0 節参照。

担当者 Name	TO-DO (今後の予定)	Doing (当日のタスク)	Suspend (待ち)	Done (完了)
AAA	…資料作成 Material creation …資料作成 Material creation …打合せ Meeting	…資料作成 Material creation …打合せ Meeting	…資料作成 Material creation	…資料作成 Material creation …打合せ Meeting
BBB	…資料作成 Material creation …打合せ Meeting	…打合せ Meeting		…資料作成 Material creation
CCC	…資料作成 Material creation …打合せ Meeting …打合せ Meeting	…資料作成 Material creation …打合せ Meeting	…打合せ Meeting	…打合せ Meeting

図 9 タスクボードの例

➤ チームマネジメント A3

- ✓ 目的
 - ✧ TOC 思考プロセスに基づき、抵抗の 6 階層に対応して、現場に納得感のある改善策に導く
 - ✧ タスクボードによる改善活動を標準化することで改善支援者の負荷を下げ、広く展開しやすくする
- ✓ 内容
 - ✧ チームマネジメント A3 は、TOC 思考プロセスに基づく改善手順であり、タスクボードで現状を認識、改善ポイントを明確化し、その根本原因を分析し、対策を計画・実行するものである。
 - ✧ その手順は 8 枚の標準フォーマット(PowerPoint スライド)であり、A3 用紙印刷で読めるレベルの簡潔さに纏められる(図 10)。A3 用紙にまとめられるため「チームマネジメント A3」と呼ぶ。各スライドの詳細は 0 節参照。
 - ✧ チームマネジメント A3 を用いたタスクボードによる業務改善は、4～8 名程度のチームによる活動を基本とし、チームから改善推進担当者とそのメンターを選出する。
 - ✧ その 2 名が中心となり、隔週 1 時間の改善支援者との面談を通じて、各スライドを作成しながら、自律的にチームの業務改善を推進する。
 - ✧ チームマネジメント A3 のフォーマットに沿って改善を検討することで、適切な改善策を導くことができるとともに、改善支援者の負荷も大幅に低減することができる。

<p>チーム名 Team name</p> <p>チームマネジメントA3 (フォーマット) Team Management A3 (Format)</p>	<p>チーム名 Team name</p> <p>① チームマネジメント運用ルール Team management rule</p> <p>ルール(飛び込みの内容) Rule</p> <p>毎曜日 9:30 までにTO DO列に貼り出す Each person begins to stick one week's worth of tasks to the TO DO column by Monday 9:30.</p> <p>週次計画 Plan</p> <p>タスクコードの色でタスクを区別する Each task is distinguished by the color of tag:</p> <p>黄: テーマ2111関係、青: テーマ672関係、赤: 飛び込み未対応 Yellow: X21, Blue: W671, Red: Interrupt tasks <タスクコードのサイズ> 75mm x 25mm (W: 75mm x H: 25mm) : 75mm x 25mm : 2hours</p> <p>タスクコードの レンジ Column</p> <p>TO DO: 今週の計画タスク Doing: 今週の実行タスク Done: 今週の終了タスク Done: 五タスク</p> <p>タスクの移動 Move of task</p> <p>毎日 9:30 ～ 4人でタスクを移動させながら報告 Sharing by visualization and the stand-up of the work by the task board</p> <p>タスクの軽減 Reduction of tasks</p> <p>毎日 9:30 ～ 4人でタスクを軽減する Moderator of the weekly rotational basis</p> <p>振り返り Reflection</p> <p>金曜日 16:00-17:00 Friday 16:00-17:00</p> <p>その他 Others</p> <p>1日で完了しなかったタスクには赤丸をつける Input a red circle to the task was not completed in one day</p>	<p>② Keep: チームマネジメント導入 工夫と成果 Keep: Team management introduction</p> <p>タスクコードを貼ったチームマネジメントを実施するので、以下のよう工夫と成果をえた(タスクコードの項目はも含む)</p> <p>Improvements and achievements in team management by task board</p> <p>実施したこと・工夫したこと Implementations & Improvements</p> <p>実施結果と成果 Results & Achievements</p> <p>タスクコードによる作業の可視化と効率化による作業時間の短縮 When the business plan, has become able to act out the work by the task board, it has become possible to shorten the working time by visualizing the work by the task board.</p> <p>タスクコードにヘンデイング欄を設置 installed suspend column to task board</p> <p>本日計画通り、各業務を順次実行して目標を達成する事が可能になった Each business has been able to be carried out in accordance with the daily plan, and the target has been achieved.</p> <p>飛び込みが減ったので各部署がスムーズに連携して、課題発見の初期解決が早くできるようになってきた As the number of interruptions has decreased, it has become possible to resolve the problem-solving pending because it has become possible to do so.</p>																																																																						
<p>③ Problem: 現状の問題分析 Problem: Problem analysis of current situation</p> <p>改善されない現状の問題 Current problem of issue</p> <p>飛び込み作業理由 Reason of interrupt tasks</p> <p>頻度(発生件数) Number of Incidents</p> <p>影響度(工数) Man-hour</p> <p>飛び込み作業理由 Reason of interrupt tasks</p> <p>頻度(発生件数) Number of Incidents</p> <p>影響度(工数) Man-hour</p> <p>作業量の見積りミス Estimates mistake of task time</p> <p>32</p> <p>69</p> <p>QAの問い合わせ Inquiry from QA</p> <p>25</p> <p>78</p> <p>不具合対応 Fault correspondence</p> <p>19</p> <p>48</p>	<p>④ 改善に取り組むテーマ Improvement theme</p> <p>[改善テーマーImprovement theme]</p> <p>営業からの問い合わせによる突発業務業務の低減 Reduction of interrupt tasks by inquiries from the sales</p> <p>[テーマ設定の理由 Reason of theme]</p> <p>XXXXXX</p> <p>31%</p>	<p>⑤ 根本原因分析 Root cause analysis</p> <p>B-Dの仮説 Reason (Assumption) -XXXXX</p> <p>必要条件 Needs B (Need1)</p> <p>行動 Action D (Action1)</p> <p>共通目的 Objective A (Objective)</p> <p>C (Need2)</p> <p>D' (Action2)</p> <p>対立 C-D'の仮説 Reason (Assumption) -XXXXX -XXXXX</p>																																																																						
<p>⑥ 対策案検討 Improvement ideas</p> <table border="1"> <thead> <tr> <th>仮定 Assumption</th> <th>対策案 Improvement ideas</th> <th>効果(失敗,成功) Effect (失敗,成功)</th> <th>難易度 Difficulty (1,2,3)</th> <th>評価 Evaluation</th> <th>備考 Remarks</th> </tr> </thead> <tbody> <tr> <td>1 XXXX</td> <td>CCPM</td> <td>2</td> <td>3</td> <td>◎</td> <td></td> </tr> <tr> <td>2 YYYY</td> <td>XDDP</td> <td>3</td> <td>1</td> <td>△</td> <td></td> </tr> <tr> <td>3 ZZZZ</td> <td>AAAAAA</td> <td>2</td> <td>2</td> <td>○</td> <td></td> </tr> <tr> <td>4 ZZZZ</td> <td>BBBBBB</td> <td>1</td> <td>2</td> <td>×</td> <td></td> </tr> </tbody> </table> <p>【取り組む改善策 Improvement】</p> <p>CCPM(Critical-Chain Project Management)</p>	仮定 Assumption	対策案 Improvement ideas	効果(失敗,成功) Effect (失敗,成功)	難易度 Difficulty (1,2,3)	評価 Evaluation	備考 Remarks	1 XXXX	CCPM	2	3	◎		2 YYYY	XDDP	3	1	△		3 ZZZZ	AAAAAA	2	2	○		4 ZZZZ	BBBBBB	1	2	×		<p>⑦ 対策案実行計画 Improvement plan</p> <p>目的 Objective X000000X</p> <p>実施 Implementation CCPM</p> <p>成功基準 Success criteria 納期の25%短縮 Delivery time 25% reduction</p> <table border="1"> <thead> <tr> <th>実施段階 Phase</th> <th>期間 Period</th> <th>実施 Implementation</th> <th>実績 Actual</th> <th>実績 Actual</th> <th>実績 Actual</th> <th>実績 Actual</th> <th>実績 Actual</th> </tr> </thead> <tbody> <tr> <td>準備 Preparation</td> <td>1月 Jan</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>実施 Implementation</td> <td>2月 Feb</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>評価 Evaluation</td> <td>3月 Mar</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>実施 Implementation</td> <td>4月 Apr</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> </tbody> </table>	実施段階 Phase	期間 Period	実施 Implementation	実績 Actual	実績 Actual	実績 Actual	実績 Actual	実績 Actual	準備 Preparation	1月 Jan	○	○	○	○	○	○	実施 Implementation	2月 Feb	○	○	○	○	○	○	評価 Evaluation	3月 Mar	○	○	○	○	○	○	実施 Implementation	4月 Apr	○	○	○	○	○	○	<p>⑧ 實施評価と横展開 Effect & Expansion</p> <p>【成果 Effect】 XXXXXXXXXXXX</p> <p>【横展開 Expansion】 YYYYYYYYYYYY</p>
仮定 Assumption	対策案 Improvement ideas	効果(失敗,成功) Effect (失敗,成功)	難易度 Difficulty (1,2,3)	評価 Evaluation	備考 Remarks																																																																			
1 XXXX	CCPM	2	3	◎																																																																				
2 YYYY	XDDP	3	1	△																																																																				
3 ZZZZ	AAAAAA	2	2	○																																																																				
4 ZZZZ	BBBBBB	1	2	×																																																																				
実施段階 Phase	期間 Period	実施 Implementation	実績 Actual	実績 Actual	実績 Actual	実績 Actual	実績 Actual																																																																	
準備 Preparation	1月 Jan	○	○	○	○	○	○																																																																	
実施 Implementation	2月 Feb	○	○	○	○	○	○																																																																	
評価 Evaluation	3月 Mar	○	○	○	○	○	○																																																																	
実施 Implementation	4月 Apr	○	○	○	○	○	○																																																																	

図 10 チームマネジメント A3 のフォーマット

4.2 全体の進め方

前節に示したタスクボードを用いたチームマネジメント A3 による業務改善プロセスは、一般的な流れの全体像は、図 11 のようにメインとなる 6 ヶ月プログラムを含めて、1 サイクルで 1~2 年間程度の活動となる。改善支援者とともにチームマネジメント活動を実行する中で、生産性向上 & リードタイム短縮に向けた継続的改善活動が定着するよう進める。

全体スケジュールは、以下のとおりである。

① 準備（期間：1～3ヶ月）

2~3回のワークショップ（3時間程度）を通じて、

- ✓ 基本的な課題と解決の方向性を共有する
 - ✓ タスクボードとチームマネジメント A3 の手法の学習する

② チームマネジメント導入（期間：6ヶ月）

タスクボードを運用しながら、チームマネジメント A3 を作成する。

A) 導入～定着（期間：前半3ヶ月程度）

タスクボードの運用を定着させ、以下の定性的効果を得る。

- ✓ 仕事内容の共有／コミュニケーションの円滑化
 - ✓ トラブルの早期発見と早期対策／相互支援
 - ✓ 仕事の計画ミス・抜け漏れ防止

B) 問題分析～対策実行（期間：後半3ヶ月程度）

タスクボードの運用の振り返りから、チームマネジメント A3 を使い

- ✓ 十分な設計時間を確保する上で重要な阻害要因の特定
 - ✓ 根本原因分析と対策および期待効果の検討・合意
 - ✓ (可能な場合) 対策の実行・成果の確認

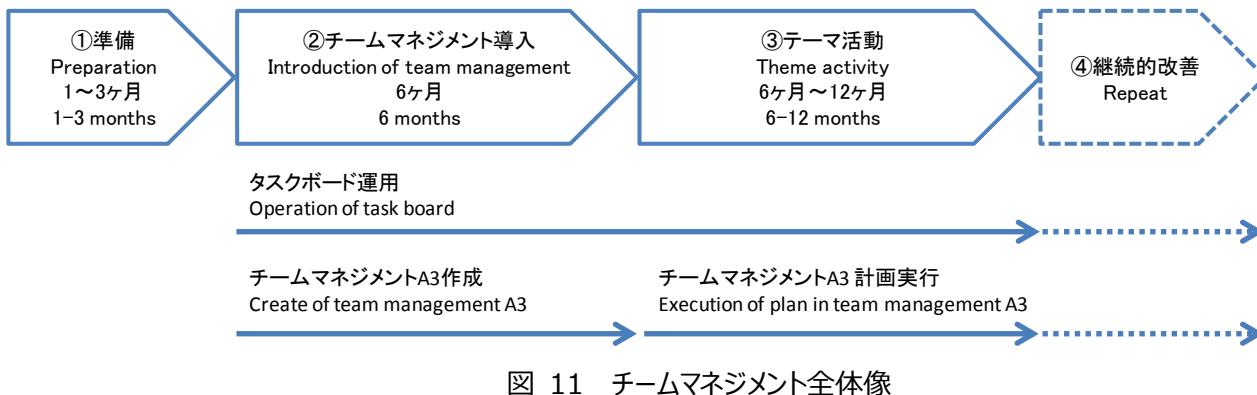
を1サイクル実行する

③ テーマ活動（期間：6ヶ月～12ヶ月）

上記のテーマ（もしくは新規テーマ）について、TOC や他の手法を適宜取り入れながら、本格的な施策実行し、定着させる。

④ 継続的改善

上記のサイクルを単独で（もしくは最小限の支援で）継続実行する。



5. 改善策の実現方法

5.1 タスクボードによるチームマネジメント運用

タスクボードによるチームマネジメントの進め方は以下となる（図 12）。

- ① 週に一度、自分自身の作業タスクを一日未満に分解して計画し、付箋に書き、タスクボードに貼り付ける。
- ② その計画に対して、作業タスクの状況をタスクボードに反映する。
- ③ 反映は、一日 15 分程度の朝会にて、付箋を動かしながら、作業状況や問題点をチーム内で共有、問題発見を迅速にし、解決を早める。
- ④ 一週間の取り組みについて、KPT(良い点:Keep, 問題点:Problem, 改善案:Try)[7]などをもちいてふりかえりを行う。

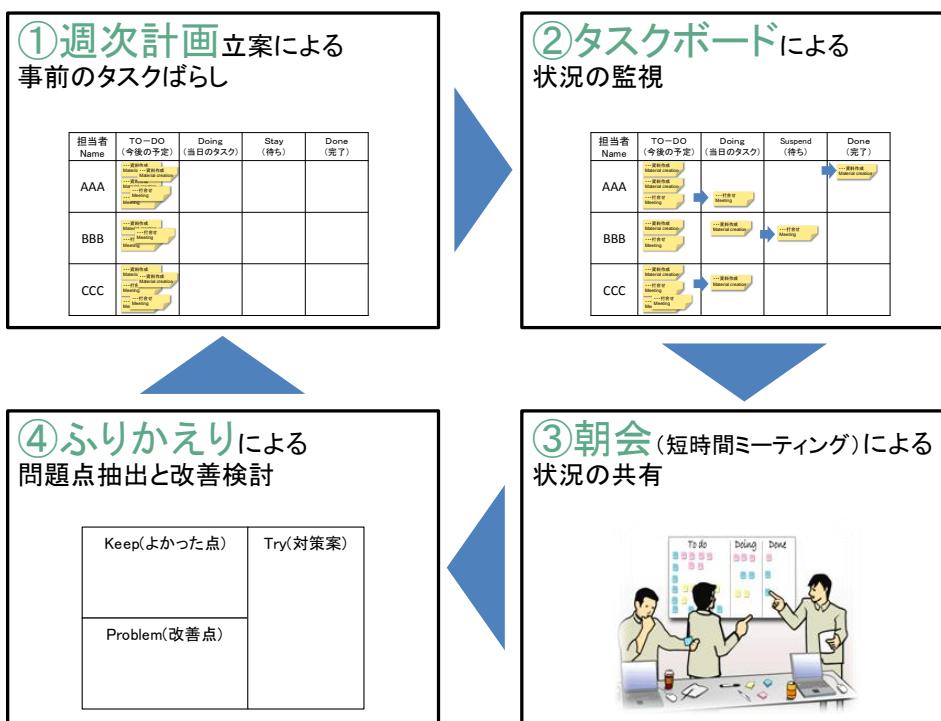


図 12 タスクボードと朝会の全体像

5.2 タスクボード上で明確にできるチームの改善ポイントの例

タスクボードには、0節に示すとおり、問題解決を早め、改善のための時間を作り出す目的の他に、チームマネジメント A3 における改善ポイントを導出するという目的がある。タスクボードにて以下のようなものが常態化している場合、改善ポイントである可能性が高い。

- ToDo の欄で、担当者によって作業量の負荷が偏っている。
→作業を他の人が手伝えないか考える。
- Doing の欄に、一日で終了しない量のタスクが貼られている。
→問題が発生している可能性あり。
- Doing から Done 欄に一日でタスクが動かない。
→何らかの問題が発生しているか、前日に優先度の高い飛び込みタスクが入った。
- Suspend(待ち)の欄に長い間タスクが待ち状態になっている。
→待ちの要因を確認する。場合によっては、問題が発生している可能性がある。

担当者 Name	TO-DO (今後の予定)	Doing (当日のタスク)	Suspend (待ち)	Done (完了)
AAA	…資料作成 Material creation …資料作成 Material creation …打合せ Meeting	…資料作成 Material creation …打合せ Meeting Meet …打合せ Meeting …打合せ Meeting	…資料作成 Material creation	…資料作成 Material creation …打合せ Meeting
BBB		…打合せ Meeting		…資料作成 Material creation
CCC	…資料作成 Material creation …打合せ Meeting …打合せ Meeting	…資料作成 Material creation …打合せ Meeting	…打合せ Meeting	…打合せ Meeting

貼り過ぎ
To many task

タスクが偏りすぎ
Task is biased

長い待ち
Long wait state

図 13 タスクボードからわかるチームの問題点

5.3 チームマネジメント A3 の各ステップ

① チームマネジメント運用ルール

チームマネジメント A3 フォーマットの 1 枚目は、タスクボードや朝会の運用ルールである。運用ルールは各チームの個別事情があるため、それぞれの事情に合わせたカスタマイズをしてもらう。

② チームマネジメント導入の工夫と成果

チームマネジメント A3 フォーマットの 2 枚目は、タスクボード運用での週次ふりかえりで出た良い点(KPT を用いている場合は Keep)をまとめたものである。また、チームごとで独自に工夫・改善した点についてもまとめる。

まずは、タスクボードによるチームマネジメント導入の定性的効果を感じることで、本格的な業務改善のモチベーションを高める。

チームマネジメント A3 のプロセスでは、改善のアプローチを 2 種類考えている。それそれ以下である。

- 解決志向アプローチ
 - ✓ すぐに対策可能であり、効果が期待できる改善アプローチ
 - ✓ 週次のふりかえりで検討し、効果を見る
 - ✓ ②に工夫と成果をまとめる
- 原因追及アプローチ
 - ✓ 簡単には対策できない根深い問題を原因追及して改善するアプローチ
 - ✓ チームマネジメント A3(③以降)で対策を検討して実行する

③ 現状の問題分析

チームマネジメント A3 フォーマットの 3 枚目は、週次ふりかえりで議論し、②の解決志向アプローチでは改善しきれない、現状の問題点(KPT を用いている場合は、Keep)を纏めたもの。できるだけ客観的にチーム内の現状を把握するために、定量的に纏める方がよい。0 節に示した、改善ポイントがハッキリするようにタスクボードからデータが取れるとよい。

④ 改善に取り組むテーマ

チームマネジメント A3 フォーマットの 4 枚目は、③の結果を受けて、チーム内で取り組む改善のテーマとその理由である。③同様、できるだけ客観的にチーム内の現状を把握するために、定量的に纏める方がよい。ただし、抵抗の 6 階層を意識し、客観的に見た場合に合理性に欠いていても、チーム内の合意形成を重視する。そのため、③とは直接関係のないテーマでも構わないが、その後の進め方は慎重に行う必要がある。

⑤ 根本原因分析

チームマネジメント A3 フォーマットの 5 枚目は、④の改善テーマに対して、その根本原因を分析したものを纏める。フォーマットは、TOC 思考プロセスで根本原因分析に用いる対立解消図[1][2][3](図 14)などを用いて行う。この対立解消図は、「改善したいけどできない」というジレンマを整理し、そこに潜んでいる仮定(Assumption)を明かにする。

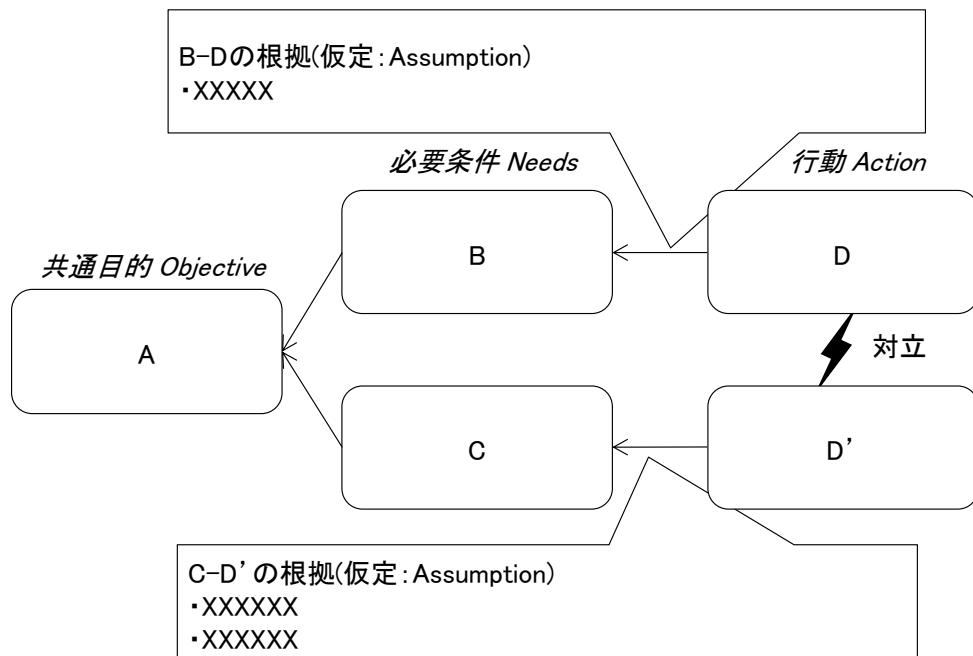


図 14 対立解消図

⑥ 対策案検討

チームマネジメント A3 フォーマットの 6 枚目は、⑤の根本原因分析に対して、考えられる対策案を列挙し、効果と難易度からどの実行する対策を決定する。⑤にて対立解消図を用いている場合、ジレンマを発生している仮定に対する対策を考えることで、ジレンマを取り除く方法を検討する。

⑦ 対策案実行計画

チームマネジメント A3 フォーマットの 7 枚目は、⑥で決定した対策案の実行計画である。まず、実行にあたり改めて、対策の目的、対策内容、成功基準を明確にする。ここでは、対策が実際に目的を達成し、成功したことを認識するために、成功基準が重要となる。ハッキリとそうであると判定できる基準でなければならず、できる限り定量的に記述するようにする。次に、その目的、対策内容、成功基準を実行する計画を期間、担当者なども含めて立案する。

⑧ 実施評価と横展開

チームマネジメント A3 フォーマットの 8 枚目は、⑦を実際に実行した結果の評価を纏めたものである。定量的なデータとして成果を纏める。また、データの収集は追加負荷のかからないものを選ぶと良い。例えば、タスクボードの付箋のカウントなどである。また、残業低減などの目的に対応する「効果」に関するデータと、対策の適用累計などの「実施」に関するデータを取るとモチベーションにも繋がるためよりよい。ただし、実際には、チームマネジメント導入の 6 ヶ月では、成果が出ていないことが多いため、「こうなればよい」という予測の特性となる。

6. 改善による変化や効果

6.1 業務改善の定性的効果について

本手法による業務改善手法の定性的効果を適用チームから得た。概ね以下に示す内容であった(主にチームマネジメント A3 フォーマット②の内容)。

1. 情報共有について

- ✓ 自分以外の担当者がどのような業務を実施しているのか分かるようになった。
- ✓ タスクボードへ貼り出して見ることにより、業務の優先順位を整理し易くなった。
- ✓ メンバー個人の作業量の偏りが把握し易くなってきた。
- ✓ 毎日実施することで、大きい項目を細分化する意識、および一つ一つ片付けて後戻りが無いようにする意識が強くなった。

2. ミス防止について

- ✓ 業務計画時、各業務を細分化して遂行時間を見積もって仕事を進める行動ができるようになってきた。
- ✓ 数週間停滞していると忘れがちであったが、相手をフォローアップできるようになったとともに、予定作業として準備できるようになった。
- ✓ 保留になっている業務をリーダーが迅速に把握して、保留原因の問題解決が早くできるようになってきた。
- ✓ 助言を得られる機会を得やすくなった。
- ✓ 業務時間を正確に見積もるため、業務を細分化して本人が意識しない問題を指摘され、問題発生前に問題を解決できた。

3. 課題抽出について

- ✓ 飛び込み作業について件数、内容が分析でき、チーム内の飛び込み業務に対する考え方を整理できた。
- ✓ 設計手配業務の不定期な割込が、業務遅延要因となっていることが判明した。手配業務は、専任者を決め、各設計者より適宜手配業務を移管することで手配業務による遅延時間は低減している。
- ✓ 図訂を先送りした事を忘れてしまい、次の作番で問題になる事がある。忘れないようにするため欄を設置して注意を促すようにした。
- ✓ 業務遅延の要因（内容、自責、他責）を把握し、発生頻度を記録することで、頻度の高い問題や遅延要因

に焦点を当てた改善活動ができた。

4. その他の成果

- ✓ 職場での会話が格段に増えた／今まであまり話さないと思っていた若手が実は結構積極的にキチンと話せることがわかった。
- ✓ 従来は打ち合わせの雰囲気が悪く・対立的な会話が顕著だったが、毎朝笑顔で情報共有し、気持ちよく仕事を開始できるようになった。
- ✓ チームワークの状況を調査するオンラインアンケートのスコアが向上した。
- ✓ 個人・家庭の事情を共有し、助け合う雰囲気が、より強くなった。時短勤務や家庭の事情による遅刻・早退等を責める雰囲気や気持ちが軽減された。
- ✓ 若手が仕事の段取りを覚えて自分で仕事をこなせるようになった。また報告や連絡の内容・タイミングが的確になってきた。

6.2 業務改善の定量的效果について

本手法を全展開した部署（18 チーム）にて、下記のような大きな定量的な効果を得ている。ただし、当該部署では本手法以外の施策にも取り組んでおり、それらの総合的な結果である。

- ✓ 製造不良時間：70%削減
- ✓ 納期遵守率：71% → 98%（内部管理指標）

また、それ以外の部署においても、下記のような定量的成果を得ている。ただし、仕事の変動による影響が大きいこと、また着手まもないチームも多く、全てのチームで定量的効果を検証できているわけではない。また、本手法に加え、並行して実施した他施策との相乗効果が含まれる可能性もある。

- ✓ 残業時間(休出含)：720 時間/月 → 480 時間/月(33%減)
- ✓ 納期遵守率：52% → 75%（内部管理指標）
- ✓ 納期遵守率：平均遅延日数を半減
- ✓ リードタイム：仕様決定までの問い合わせ数 20%削減による期間短縮
- ✓ リードタイム：5 日 → 1 日（仕様変更対応日数）
- ✓ クレーム件数：6 件/案件数 70 件 → 0 件/案件数 135 件
- ✓ 生産性：1.5 倍（対応案件数）※人員および残業の増加無しに達成
- ✓ チームワークに関するアンケート³：10 点/10 点満点 ※平均は 5～7 点程度。

6.3 改善活動の展開について

本手法による業務改善を取り組んでいるチーム数を

³ OCAPI「組織のメンバーの関係性、課題に対する考え方、変化を生み出す行動などの定性的指標を数値化するアンケート」

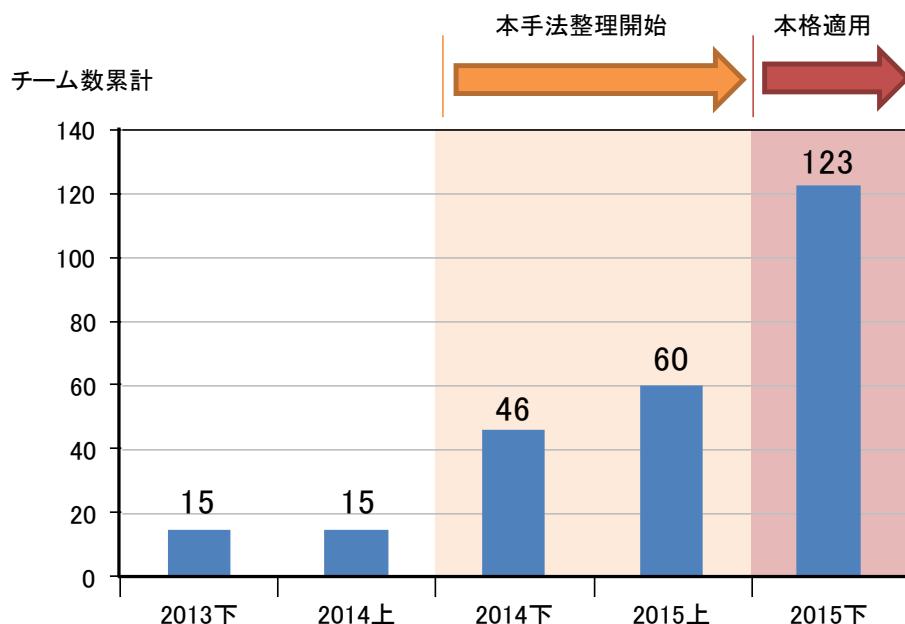


図 15 に示す. 本手法は, 2014 年下期から検討が開始, 試行を経て, 2015 年下期から本格的に展開を開始した.

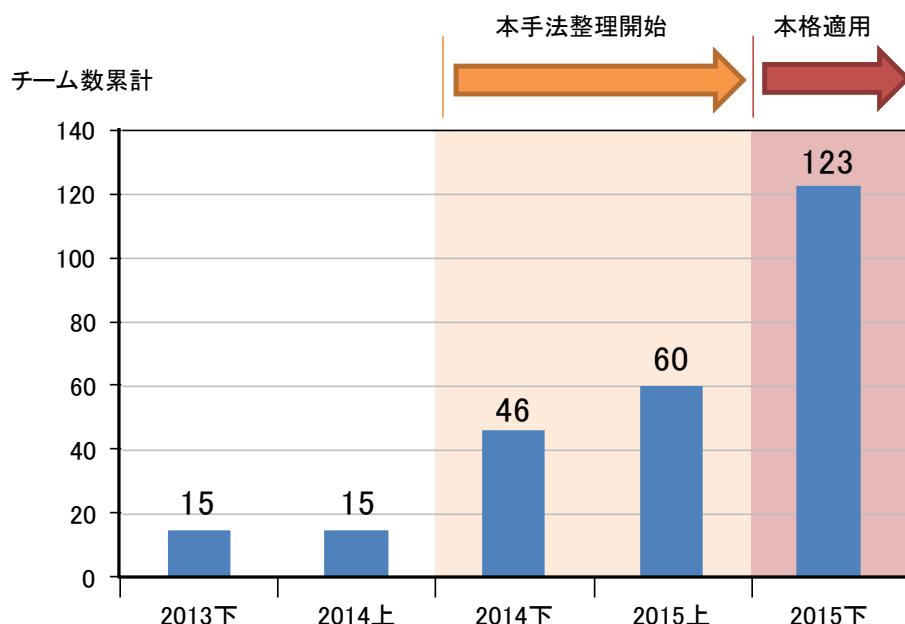
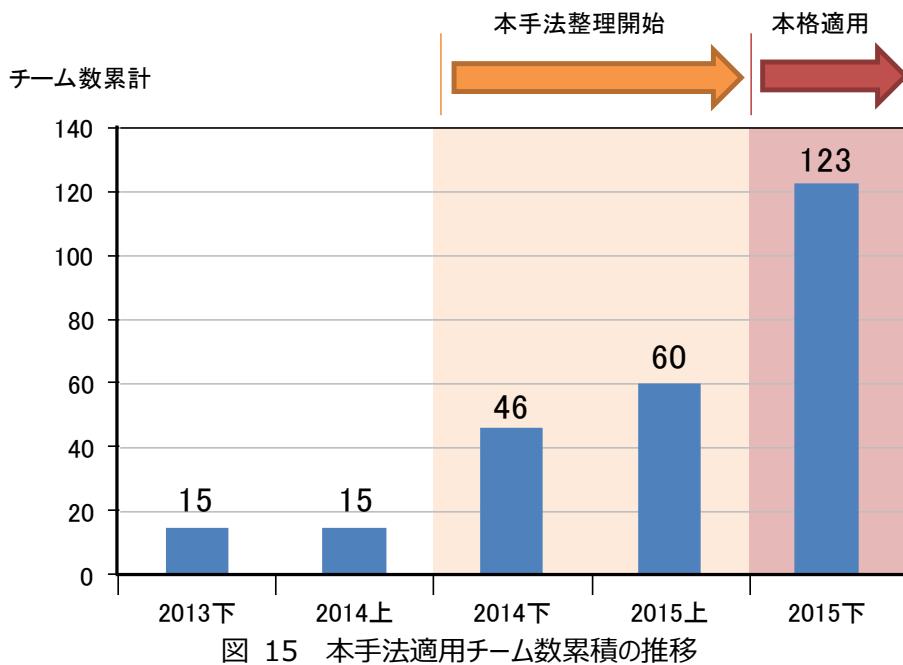


図 15 に示すとおり, 口コミでの評判も手伝って, 2015 年下期から取組みチーム数が倍増している (現時点で, のべ約 1,000 名が関わっている). 展開規模は大きく倍増しているが, その間の改善支援者であるコンサルタントのリソースは, 3~4 名と大きくは変動していない. そのため, 本手法は, 標準化したことによって, コンサルタントの負荷を増やさずに, 効率的に業務改善を進めることができると考えられる.



7.改善活動の妥当性確認

本報告では、効果的かつ効率的な業務改善の方法として、タスクボードを用いたチームマネジメント A3 による業務改善プロセスを提案した。本手法は、タスクボードと朝会を用いて、チーム内の作業タスクの状況を共有し、問題解決を早めることで、改善のための時間を作り出すことができるため、現場ではじめやすく、チームの改善推進担当者が、TOC 思考プロセスをタスクボード＆朝会向けに標準化したフォーマット「チームマネジメント A3」を作成しながら、タスクボードによる業務改善を行うことで、効果的な業務改善を実行することができる。

結果、「メンバー個人の作業量の偏りが把握しやすくなった」「タスクを一つずつ片付けることで後戻りを減らすようになった」「助言を得やすくなった」などの定性的効果に加え、残業時間 33% 減、製造不良時間 70% 減、納期遵守率 27% 増、などの定量的効果もあがっている。

また、チームマネジメント A3 という標準フォーマットを用いることで、業務改善を指導するコンサルタントの負荷も減り、同様の人員リソースで改善活動を行うチーム数を倍増することに成功した。結果、2016 年現在で、日立グループのべ 1,000 名の規模で業務改善が広がっている。

A. 参考文献

- [1] エリヤフ・ゴールドラット, ザ・ゴール 2 —思考プロセス, ダイアモンド社, 東京, (2002).
- [2] 村上悟, 問題解決を「見える化」する本, 中経出版, 東京, (2008).
- [3] 岸良裕司, 全体最適の問題解決入門—「木を見て森も見る」最強の思考プロセス—.
- [4] エリヤフ・ゴールドラット, ザ・ゴール—企業の究極の目的とは何か, ダイアモンド社, 東京, (2001).
- [5] “プロジェクト・ファシリテーション,” <http://www.objectclub.jp/community/pf/>
- [6] 西原隆, 栗山潤, TOC/CCPM 標準ハンドブック—クリティカルチェーン・プロジェクトマネジメント入門, 秀和システム, (2010).
- [7] 天野勝, これだけ ! KPT, すばる舎, (2013).

2B4 「永遠の目標「ヒューマンエラー ゼロ」達成の秘訣」 渡辺聰美(富士通エフ・アイ・ピー)

<タイトル>

永遠の目標「ヒューマンエラー ゼロ」達成の秘訣

<サブタイトル>

組織全員で取組む“意欲向上”活動が壁を破る

<発表者>

氏名（ふりがな）：渡辺 聰美（わたなべ さとみ）

所属：富士通エフ・アイ・ピー株式会社 サービスビジネス本部 センターサービス統括部
明石 DC オペレーション部

<共同執筆者>

無し

<要旨・主張したい点>

ヒューマンエラー削減に向け、従来はITサービスに関する各種標準の導入や医療、航空、交通業界に学ぶ人間重視のプロセス整備を進めてきた。しかし、このように十分に考慮された仕組みを導入しても、ヒューマンエラー撲滅には至らなかった。

本発表では、当社運用部門のヒューマンエラー撲滅に繋がった効果的なアプローチ事例として「組織全員で取組む“意欲向上”活動の作り方」を紹介する。意欲向上は仕組み作りとは車の両輪のような関係で、どちらが欠けても効果は期待できないと考えられる。さらには意欲向上に取組むことで、品質向上に加え、セキュリティ確保という効果をもたらすことも併せて紹介したい。

<キーワード>

ヒューマンエラー、ヒューマンエラー削減、ヒューマンエラー撲滅、意欲減少、動機づけ、EQ理論、人材育成、コミュニケーション、再発防止策、ダブルチェック、改善提案、意欲向上、合意形成、モチベーション、テーラリング、情報漏えい防止

<想定する聴衆>

高成熟度組織の方、品質管理・プロセス改善・人材育成に携わる方、組織長、プロジェクトマネージャー、プロジェクトリーダー

<活動時期>

2014年度 現状分析～改善～評価

2015年度 展開～維持

2016年度 維持

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

データセンターの重要性は年々高まり、近年では、新しい時代の「社会の重要インフラ」ともいえる存在となった。データセンターに対する期待の中心はセキュリティ対策や災害対策だが、暗黙の期待に「サービスを止めない」「情報を漏らさない」というような要件が含まれることはいうまでもない。

一方、システムトラブルの原因で最も多いのは「ヒューマンエラー」である。当社では安定稼働のため、ヒューマンエラー削減に様々な視点から取組んできた。具体的にはITサービスに関する“標準”的導入、ヒューマンエラー発生防止に関する研究が進んでいる医療、航空、鉄道等業界のナレッジの活用である。

その結果、ヒューマンエラーは撲滅には至らないものの、低いレベルで推移し、人間の限界を考えると、一定の成果を挙げたとも思われた。しかしある年、前年比2倍のヒューマンエラーが発生し、従来の取組みの見直しを余儀なくされた。

2.改善したいこと

ヒューマンエラー削減に向けた取り組み状況の振り返りを行ったところ、再発防止、未然防止のそれぞれに以下のような問題を発見した。

<再発防止>

- ・再発が防止されているケースを現場にて精査したところ、再発防止策が定着していないケースがしばしば検出された。
- ・効果的と思われる対策が現場に定着しない理由について、現場にインタビューしたところ、コミュニケーションが停滞していると思われるような様子が感じられた。

<未然防止>

- ・類似障害事例は他人事であり、自身は大丈夫と思っているケースも多い。（起こさないように注意しようという意識不足）
- ・異常を検知する力が低下している。

これらの問題を解決することで、ヒューマンエラー撲滅を実現したいと考えた。

3.改善策を導き出した経緯

前述の問題に対し、原因分析を行ったところ、いずれのケースにも現場要員の意欲減少、人間力の低下というような要素が関連していることが垣間見えてきた。さらに現場要員のインタビューを重ねていくと、役割間の力関係、上下関係といった“壁”が意欲減少に関与していることが推察された。その“壁”を破るために、専門家の知見を参考とした。

いわゆる動機づけの研究については、マズローの「欲求五段階説」やハーズバーグの「動機づけ衛生理論」が知られている。今回はこれらの中でも「他者からの評価」、特に人事システムや上司からの評価ではなく、現場で働く仲間からの評価に着目した。しかし、本来動機は自分で付けるものであり、他者から付けられるものではない。そこで、システム運用現場の特性に目を向けた。システム運用現場は、安定性を重視するあまり、従来の方法を“変える”ことに消極的となったり、ミスが起きると人的介入を強化しがちな環境である。このことが、個々人に自身の成長や貢献を感じられる機会を与えることを困難なものとしていた。

そこで、日常の業務の中で一見、何でもないような問題、皆が見過ごしてしまいがちな問題に目を付け、一つの課題としていく、そのような文化を一人ひとりに醸成したいと考えた。最大のヒントを得たのは、EQ理論である。そのような“気付き”的には感情のコントロールが重要であり、日常の言動や行動を変え、マイナス方向に左右しがちな感情をプラス方向に変えることで、人の行動に影響をもたらすことが出来るという考えだ。このようにして、職場の仲間との間にプラス方向の感情を多く芽生えさせる組織活動への変革を行うこととなった。

4.改善策の内容

1) トップダウンで壁を破る（品質スローガンの変革）

従来の品質スローガンは安定稼働の方針、目標そのものであった。その方針、数値目標によって現場一人ひとりの意

識向上を図ることは困難であり、そこには、トップと現場の間に“壁”があるような状態であった。そこで、従来の品質方針の体系を見直すこととした。

見直されたスローガンは「一人ひとりで作る品質」。“気付く力”を高めること、相手を“気遣う対応”を心がけるという2つのシンプルなメッセージで構成した。このメッセージの根底には、“気が付かない人”なんていいないという考え方があった。皆、気付く力を持っている。一見、気が付かない人と見えたとしても、その障壁さえ破れば、誰もが“気付き”を発揮できるというものである。そして、その先に自ずと安定稼働がついてくると考えた。

2) 苦労を伴う活動からの脱却（人材育成の変革）

システム運用の現場は、安定稼働のため、教育、周知を重視しており、実施される頻度も日常的に多くなる傾向がある。現場担当者に業務の重要性を認識させるため、厳しく諭すシーンが多くなっていた。このことは、現場担当者への実質的な負担に加え、教える側と教える側の間に“壁”があるような状態であった。そこで、まずは、最も現場に負担のかかっている新規要員向け教育を見直すこととした。

見直しの方針は「学びたいという意欲を醸成する場」への変革とし、以下の2点を特に意識した。

- ・“教える”から“気付きを与える”に変革する
- ・いきなり“厳しい”から入り口は“楽しい”に変革する

教育の場をこのように変革するために、教育で取り上げる題材を難しい業務関連素材ばかりではなく、入り口で接する題材を日常生活の中にあるようなものに変えた。この考え方の裏には、日常生活においては、誰もが“改善”的な実践者であるというものがあった。日常生活では、自然と発揮できている能力が、会社での業務になった途端、なかなか発揮されていないケースを克服したいと考えた。

5.改善策の実現方法

1) 品質スローガンの浸透

相手を気遣う対応について、一人ひとりが日常の業務シーンで意識できるようにするために、ポスターやポケットカードを作成し、各自の好みの場所に掲示してもらった。もちろん掲示の強制はせず、カードは、温かみを感じさせられるよう、イラストも併用した。

2) 教育の変革

- ・“教える”から“気付きを与える”に変革（形式の変更。知識重視からマインド重視へ）

伝えるべき知識は数えきれないほど存在する。けれども、新規要員向け教育で全てを伝えても消化は困難であるという実態を冷静に評価した。講師から受講生に伝えることが主体、いわば一方通行の講義形式という従来方式を、受講生との対話をベースとした双方向講義形式、演習形式に改めた。このようにすることで受講生の認識度合いを都度確認でき、さりげない“賞賛”（褒め言葉）を提示する機会とすることことができた。

さらには、受講者の短所については、否定することなく、一歩克服することに向けたアドバイスを行うことで、長所の更なる伸長と短所克服の意欲向上をもたらした。

- ・いきなり“厳しい”から入り口は“楽しい”に変革（品質マインドの醸成後、知識習得の段階へ）

大量の資料をテキストとして提示され、連日続く教育。学ぶ意欲を持っている人でも、その意欲の持続が困難とも思える状況であった。大量なテキストには過去の経験が散りばめられていたものの、実際の現場では、過去の経験がそのまま使えるケースよりも、過去の経験の応用力を必要とするケースの方が多いという声もあり、刷新が必要となつた。応用力を養うということは、すなわち個人の気付く力を養うことに等しい。そのため演習素材に日常体験を盛り込むこととした。

<演習素材（一例）>

-何が見える？（トリックアート）

- どう捉える？（携帯を落とした）
- 線の上にマルを書こう
- 大きい「くも」を書こう
- 伝達（報・連・相）でうまく伝わらなかつたケース
- あなたが不満をもつたサービス、感動を受けたサービス
- あなたがネガティブな思いになつた他人の言動？（表情、態度、しぐさ、口調、言葉…）
- 自宅玄関の鍵を無くした！どう対策する？
- あなたが繰り返してしまつ失敗、繰り返さなくなつた（克服した）失敗から学ぶ「失敗回避の秘訣」

これらの素材に通ずることは「どの回答（感じ方、見え方、捉え方）にも“誤り”はなく、一人ひとりの見方が異なる」ということを体感してもらうということである。このようなメッセージを伝えられるような素材を随所に用意するという、いわば素材探しに苦慮した。主題となるテーマを提示する際、関連する副題を効果的な導入として活用することが主題に対する認識や理解向上にも繋がるため、演習素材の選定は十分に吟味したいところである。

6.改善による変化や効果

前述のような改善の結果、永遠の目標とも捉えてきた「ヒューマンエラー撲滅」が実現した。また、それに至る過程として問題視した、再発防止、未然防止の活動については、以下のような変化が見られた。これらの変化は一見、“当たり前”的なことと映るかもしれないが、現場で当たり前のように回り続ける状態まで持っていくまでは実は一苦労がある。順調に回り始めれば、阻害要因が悪影響を及ぼさない限り回っていくので、回り始めるまでのところに配慮が必要だと思われる。

また、再発防止、未然防止に繋がつたという断定は困難だが、現場起案で様々な有益な活動が開始され、能動的に継続されていることも、これらを支える土壌となつていると推察される。

＜再発防止＞

- ・再発が防止されているケースを現場にて精査したところ、再発防止策が定着していないケースがしばしば検出された。
- ・効果的と思われる対策が現場に定着しない理由について、現場にインタビューしたところ、コミュニケーションが停滞していると思われるような様子が感じられた。

⇒再発防止策が定着。コミュニケーションが活性化した。

- 再発防止策立案にあたり、現場に配慮するようになった。（特に安易なダブルチェックの導入は無くなつた。インタビュー時にミス当事者の気持ちに配慮したインタビューが実施されるようになった。）
- 再発防止策立案時、現場とのコミュニケーションを重視し、少数意見も軽視せず、現場が納得できるような議論を経るようになった。
- 再発防止策の評価を確実に実施し、必要に応じ対策を見直す（改める）ようになった。

＜未然防止＞

- ・類似障害事例は他人事であり、自身は大丈夫と思っているケースも多い。（起こさないように注意しようという意識不足）
- ・異常を検知する力が低下している。

⇒改善提案が活性化した。（件数ベースで 20%UP）

- 件数の目標化ではなく、解決までのスピードを目標とするように変化し、改善意欲向上に繋がつた。

- 改善提案者への賞賛に加え、改善実施者への“貢献”を賞賛するようになった。（気付きと“汗”を評価）
- 効率化提案の裏で品質向上が阻害されないか、現場一人ひとりが考えて提案するようになった。
- 関係者一体となって、ミスを発生させないこと、若しくは発生しても、いち早く検出したり、影響を出さないことについて、知恵を出す文化が醸成された。（自身の業務テリトリーに関する提案から、社として提供しているサービス全体に対する改善提案を生み出そうという考え方へ変わった。）

＜現場から生まれた様々な活動より代表事例＞

・コンセンサスゲーム

コンセンサスゲームとはチームメンバーとの合意形成（コンセンサス）を行う必要があるゲームとして、ネットでも紹介されている。当社では、オペレータ自らが企画、定期開催し、現在では、8シリーズ目を数える。その結果、互いのアイデアを尊重しあう文化が醸成され、コミュニケーションが活性化し、報告・連絡・相談も効果的に実施されるようになった。

・部門間交流

部門間の活動を現地にて互いに共有しあう活動である。指摘はせず、伝えるだけ。良いところを持ち帰る活動と定義している。持ち帰るかどうかも持ち帰る部門が決定権を持つという、自らの意思をベースとした横展開活動である。紹介部門の中では、当たり前と考えながら実施してきたことも、他部門から見ると素晴らしい工夫が多く存在した。褒められることによるモチベーションアップに加え、持ち帰る部門にとっても、自部門のニーズ、体力に応じ、改善計画を立てることが出来、効率的な品質向上が可能となった。

改善にあたり、多くの組織では他社、他部門事例等、様々な成功事例を参考し、取り入れている。しかし、効果を生み出すためには、取り入れ方に工夫が必要と考える。つまり、模倣だけでは、十分な効果に至らないといえる。現場が心から継続したいと考えていれば、模倣を経て、自らにあった方法、テーマを工夫し、さらには継続しようと試み始める（いわゆるテラリング）。現場は、自分たちで楽しみながら、活動を作り、管理者はそんな活動を支援する。そのサイクルによって、効果は生み出される。

7.改善活動の妥当性確認

今回の改善活動を機に、より高い品質を実現するためには、品質を管理する仕組みの整備に加え、“意欲”をコントロールすることがその効果を大きく左右するという気付きを得た。

冒頭にも述べたように、データセンターの重要性は年々高まっている。本発表では、品質を中心に述べたが、もう一つの重要な柱は「お客様情報を守る（情報漏えい防止）」ということである。そこで、セキュリティ関連の問題発生状況を確認した所、情報漏えい問題も発生していないことが確認された。

一見、方向性の異なるテーマのようにも見えるが、実は情報漏えい、不祥事というような各種問題の防止にも“意欲”が影響していることが各界で述べられており、今回の活動は当社の背景に対し、妥当な改善策であったと評価できる。

品質、セキュリティ何れにしても、問題が起きてしまった場合の損失は測り知れないほど巨額なものとなり得る。仕組みの導入には一定のコストが必要となるが、意欲向上に必要なコストの主要部分は、マインド醸成にかかる初期教育である。教育コストを要しても、その後の改善等が実現されることで回収も可能となる。これらを考慮すると、意欲向上のための改善コストの費用対効果は高いと推察される。

今後も当事例を参考にして、現場リーダの育成をスピードアップし、全社での品質向上、セキュリティ確保に繋げたい。

<タイトル>

複数部門にまたがったシステム・ソフトウェア開発プロセス改善の取り組み

<サブタイトル>

～同じところ・違うところを組み合わせて、みんなで進めるみんなの改善～

<発表者>

氏名（ふりがな）：藤山晃治（ふじやま こうじ）

所属：パナソニック（株）オートモティブ＆インダストリアルシステムズ社 技術本部 プラットフォーム開発センター システム技術開発部 機能安全推進課

<共同執筆者>

氏名（ふりがな）：

所属：

<要旨・主張したい点>

システム製品の開発や事業が複数の部門にまたがる場合、一部門のシステム・ソフトウェア開発プロセス改善を推進しても、その効果が限定的になってしまう可能性がある。発表では、一部門のシステム・ソフトウェア開発プロセス改善を端として、システム開発に関連する複数の部門を巻き込んだ部門間連携を軸にし、さらに関連部門の組織特性も考慮し改善を進めることで、システム製品の品質向上を目指す取り組みを紹介する。

<キーワード>

システム・ソフトウェア開発プロセス改善、部署間連携、問題解決管理

<想定する聴衆>

ソフトウェア開発プロセス改善推進者、SPI

<活動時期>

2015年4月～継続中

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

システム製品を開発し製品化する場合、そこに関わる一部の組織のプロセス改善を行ったとしても製品としての高品質化は困難であり、関係部署を連携させながら全体としての改善取り組みを推進していかなければならない。しかし、その製品開発の関わり方も部署のミッションによってさまざまであり、それぞれが高品質な製品を提供したいという思いは共通しているものの、各組織の特徴を考慮せず均質な改善を推進しても期待する効果は得られない。さらに、最近ではUL1998に代表されるようなソフトウェアへの機能安全的な顧客要求も増えており、今回、中心となる事業体のシステム・ソフトウェア開発プロセス改善を取り組み起點とし、そこから徐々にスケープを広げながら、関係部署の開発物やミッションの違いに基づいて全体視点と個別視点の両方の面で、部門間連携による足並みを揃えた改善を進めていくことが重要となった。

2.改善したいこと

事業部単独でのSW不具合低減撲滅と、組織開発プロセスの整備、さらに派生した部門の特性を考慮した複数部門間での最低限の活動統一化による部門間連携を軸にした開発効率改善(開発プロセス・開発標準)

3.改善策を導き出した経緯

今回の事例ではシステム製品開発に関わる3部門を対象としたプロセス改善を図る。今回の部門では組織改変が続いたため、全社規定に準拠した体系的なシステム・ソフトウェア開発規定が十分ではない、という共通した課題があった。

- ① 組織Aは、事業部としてシステム製品を製品化。本活動の中心的存在。起点は、この部門の体系的なソフトウェア開発プロセスの構築であった。ただし現時点でソフトウェアにフォーカスしているため、システム全体のプロセス改善が必要。
- ② 組織Bは、組織Aのシステム製品に搭載するコンポーネントを開発しているだけでなく、開発コンポーネントを統合してシステム化して、組織Aへ引き渡すまでを行う。そのため、システム全体を俯瞰しての開発活動で先行。組織Aに引渡すシステムの品質確保が不可欠。
- ③ 組織Cでは、新規事業開発をミッションにした組織であり、その提案先が組織Aである。新規事業開発では、顧客からの要望で試作を提供することがあるが、その試作がそのまま市場に出てしまうケースもある。従って、試作の段階から出来る限り、開発規定に従った高品質な開発を行っていきたい、また事業部引渡し時の品質的な課題をできるだけなくしておきたい、という意図あり。

従って、必要な改善策を以下4のように設定した。

4.改善策の内容

①全社規定に準拠したSW・システム開発プロセスの構築

組織Aが中心に行い、その成果物を組織B、Cも活用できるようにする。

②SW・システム開発プロセス改善の推進

現行PRJでの問題解決管理プロセスの導入と、開発ツールの共通化

③個々の課題への対応

例えば、組織Cに関しては、社内の別の新規商品開発センターとの橋渡しを行う

5.改善策の実現方法

①週1回の3部門巡回(個別対応のためのウィークリー・ファシリテーション)

毎週各部門を巡回し、個別の課題や問い合わせ事項に対応。

②月一回の3部門合同MTGの開催(全体対応のためのマンスリー・ファシリテーション)

毎月3部門の関係者が集まり、改善取り組みの進捗報告や課題共有、共通ツールの活用法を議論。

③技術セミナー実施(組織力アップのためのスキルアップ・ファシリテーション)

3部門の技術者を対象にしたプロセス改善に関する技術セミナーの定期開催。

④ 機能安全対応事項のガイドライン化（将来の備えのためのフューチャー・ファシリテーション）

別事業所で展開されている車載機能安全活動のエッセンスをガイドラインへ反映。ISO 規格の説明や解説。

6.改善による変化や効果

技術者が個々に不具合対応をしていたころから比べ、プロセス準拠により不具合を適切に処理する、そして埋め込まないという意識が芽生えてきた。また、それが他部門へも良い影響を与え、同じところ・違うところを意識しながら高品質を実現しようという意識が芽生えてきた。

7.改善活動の妥当性確認

活動が軌道に乗り出してからようやく半年であるため、今後は現場技術者や SQA にもヒアリングをしながら確認をしていく。また、問題解決管理プロセス実施で得られた不具合情報の分析を通して、不具合削減に貢献しているかを確認する。

A.参考文献

2C2 「通信機器組込み用ソフトウェアの開発プロセス改善による品質向上の一実施例」 嘉屋良洋(アンリツネットワークス)

<タイトル>

通信機器組込み用ソフトウェアの開発プロセス改善による品質向上の一実施例

<サブタイトル>

上流工程は XDDP で、下流工程は総合テスト強化とテスト自動化でプロセス改善

<発表者>

氏名 (ふりがな) : 嘉屋 良洋 (かや よしひろ)

所属 : アンリツネットワークス株式会社 開発部

<共同執筆者>

氏名 (ふりがな) : 森中 秀明 (もりなか ひであき) 、村山 知寛 (むらやま ともひろ) 、東山 満 (ひがしやま みつる)

所属 : 同 品質保証部、 開発部

<要旨・主張したい点>

弊社はネットワーク通信機器を製造・販売する企業である。社内検証で合格した更新ソフトウェアの不具合がユーザ先で発生したのをきっかけとして、「客先信頼を勝ち取る体制確立」が経営課題となった。原因・対策を分析した結果、①開発工程の作り込み品質と検証体制の改善、②妥当性確認のプロセス改善が必要と判断した。①について、SEPG を新たに組織し、機能設計では XDDP を導入した。テストエンジニアチーム (TE チーム) を創設し、総合テストの質量の充実と、リグレッション・テストの自動化を行った結果、一定の成果を得られたので報告する。

<キーワード>

SEPG、プロセス改善、品質改善、XDDP、組込みソフトウェア、検証、機能設計、総合テスト、テスト自動化、ウォーターフォールモデル、V 字モデル、W 字モデル

<想定する聴衆>

ソフトウェア開発エンジニア、組込みソフトウェアエンジニア、SEPG、品質保証エンジニア、テストエンジニア

<活動時期>

2015 年 4 月～継続中

<活動状況>

- 着想の段階 (アイデア・構想の発表)
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他 ()

<発表内容>

1.背景

弊社の製品であるネットワーク機器や遠隔監視システムのソフトウェア開発では、既存のソフトウェアを流用し、新たな機能を追加する場合が多い。ソフトウェア機能を追加すると、後工程で既存の機能（本来の機能追加に直接関係ない機能）が動作しなくなるデグレードが発生しており、「客先信頼を勝ち取る体制確立」のためのソフトウェア品質改善が必要となった。一方、グローバル市場への製品展開を図るという経営方針から、「さらに信頼性の高い製品の開発体制を確立させる」ことが経営課題となった。これらを受けて、開発部と品質保証部からなる「ソフトウェア開発プロセス改革」チームを発足させ、①開発工程の作り込み品質と検証体制の改善、②妥当性確認のプロセス改善に取り組むこととした。本発表では、このうち、開発部所属のSEPGチームとテストエンジニアチーム（TEチーム）が主体となって取り組んだ①について報告する。開発部の設計者から分離独立させ、新たに組織されたSEPGは、機能設計にはXDDPを導入し、テストエンジニアチームは総合テスト設計と総合テスト実施、リグレッション・テストの自動化を行った。

2.改善したいこと

【改善前の問題】

弊社はウォーターフォールモデル（V字モデル）に基づいたソフトウェア開発を実施している。開発部の設計者が基本設計、機能設計、構造設計、詳細設計、コーディング、単体、結合テスト、総合テストの各工程を上流工程から順番に実施していた。また、品質保証部では、妥当性確認のためのシステムテスト工程（最終工程）を実施し、製品を出荷していた。

上流工程である機能設計で機能追加に伴う不具合を特定できず、後工程のテスト工程では不具合を摘出し切れず、その結果、市場への不具合流出につながっていた。

【改善したいこと】

従って、機能設計で機能追加に伴う不具合を作りこむ可能性のある箇所を特定し、後工程のテスト工程で不具合を漏れなく摘出するという作業を、与えられた時間の中で精度よく実行する体制とプロセスを構築し、市場への不具合流出を防止したかった。

3.改善策を導き出した経緯

- ① 機能追加やバグの修正を実施する際、機能設計において変更による影響範囲が不明確であり、変更による影響箇所の修正漏れによるデグレードが発生していた。デグレード防止の観点からのレビューを行うことが必要であると考え変更の影響範囲を見える化するための共通設計手法として「XDDP(eXtreme Derivative Development Process)」を導入することになった。
- ② 従来は、総合テスト設計と総合テストを設計者自身が行っており、設計から総合テストまで同一の設計者が担当していた。そのため、自身が変更した箇所に関する総合テストは見通せるので実行するが、変更により影響を受ける範囲を充分に特定できず、追加機能に関係ない不具合が後工程（システムテスト工程）に流出していた。そこで、設計とは分離独立した専任チームであるテストエンジニアチームにより、第三者視点で総合テスト設計と総合テストを実施できるようにした。
- ③ 機能追加やバグの修正を実施する際、デグレードを防止するために既存機能の動作確認を行うリグレッション・テストを手動で行っていたが、手動によるリグレッション・テストでは、費用と時間的制約からテスト範囲が限定される問題があった。この問題を解決するために、リグレッション・テストを自動化するテスト環境を構築した。

4.改善策の内容

V字モデルの作り込み工程である機能設計と、それに対応する総合テスト工程を同時に改善することで、品質改善効果が高まった。

①【XDDP 適用】

作り込み工程である機能設計の品質改善としてXDDPを適用し、変更前後の仕様の見える化、共通言語化を図った。XDDPの3点セットの成果物に対して、変更の内容とその影響をレビューし、上流工程での不具合の摘出を強化する一連のルーチンが確立した。

②【総合テスト強化】

テストエンジニアチームが、総合テスト設計と実機での総合テストを行い、総合テスト設計通りに実機が動作していることを確認するようにした。

機能仕様書とXDDPの変更要求仕様書とトレーサビリティ・マトリクスを作成後、すぐに設計工程と並行してテストエンジニアチームにより総合テスト設計書を作成するようにし、W字モデルとした。

第三者的な視点によるテスト項目の選定により、変更による影響範囲を広げた総合テストが実現し、網羅性が向上した。

③【レビュー強化】

上流工程で総合テスト設計を行うことにより、テストエンジニアが、テスト視点で機能仕様をレビューし、機能仕様の変更による影響範囲の考慮漏れ、誤りを防止することも出来た。さらに、総合テスト設計書について、設計者と品質保証部員がレビューすることで、より多くのメンバーが異なる観点からレビューすることになり、テスト項目の網羅性が向上し、総合テスト漏れ防止に効果があった。

設計者とテストエンジニアチームがお互いの成果物に対して、双方のレビューを行うことが、品質改善につながった。

④【テスト自動化】

優先度の高いリグレッション・テスト項目を選定し、設計者、SEPG、テストエンジニア、品質保証部員がレビューし、合意を得た後に、リグレッション・テストを自動化することで、投資対効果を高めた。

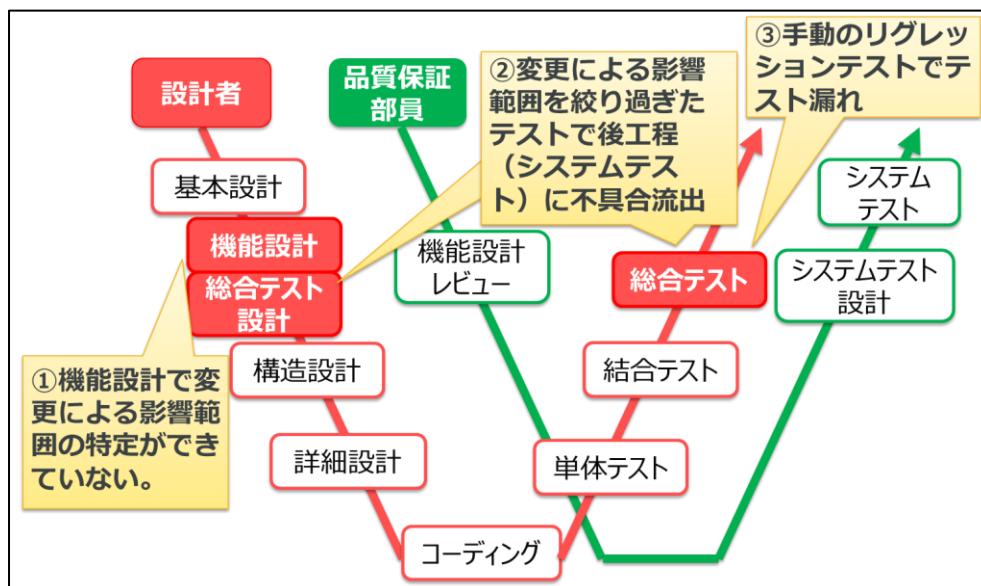
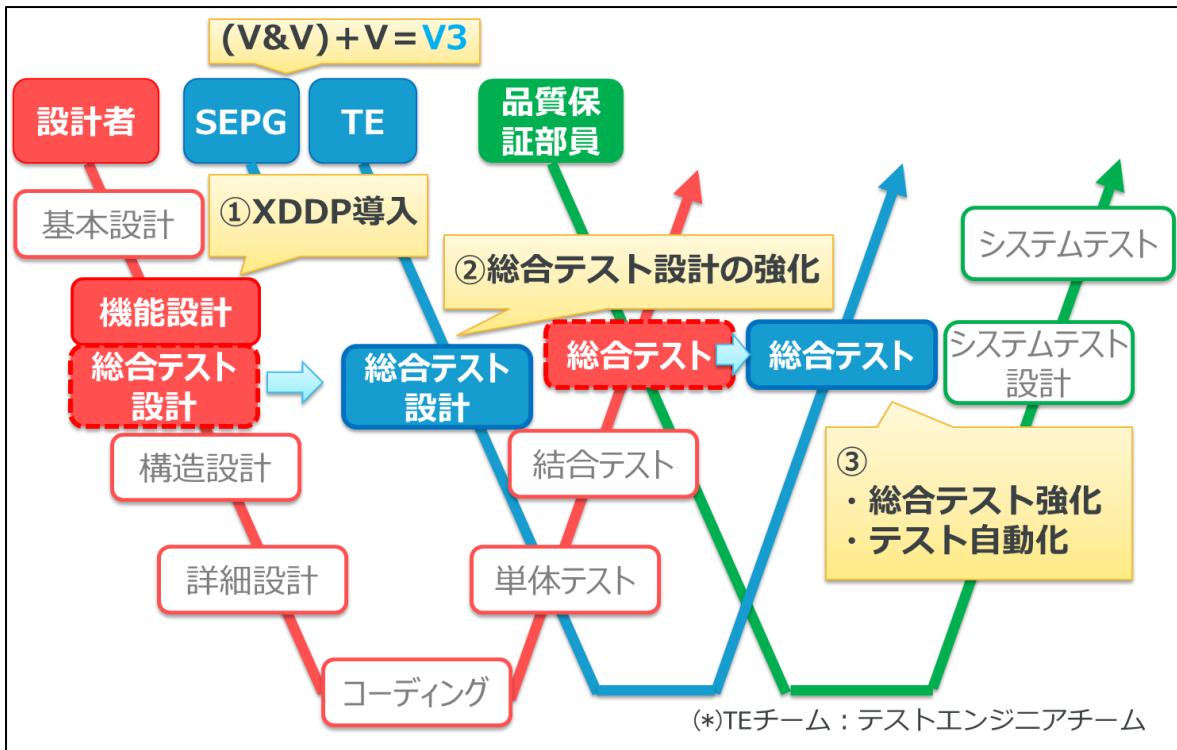


図1. プロセス改善前のソフトウェア開発



5.改善策の実現方法

① 【XDDP 適用】

弊社の開発部の設計者、SEPG、テストエンジニア、品質保証部員は XDDP に関する知識や適用経験がなかったので、最初に教育講座で、XDDP の基本を学習させた。

実際のプロジェクトに適用する場合には、最初から大規模なプロジェクトに適用すると、障壁が高く、組織になじまないリスクがあったので、スマールスタートで小さなプロジェクトから適用を開始し、徐々にプロジェクトの適用を拡大させた。また、通信制御のプロセスを扱う特徴から、XDDP に図表によるモデル化の技法を取り入れて、見える化の改善を図った。

② 【総合テスト強化】

「総合テスト設計書」の記載項目が製品により不統一だったので、統一を図った。

「総合テスト設計書」では、テストの目的、方針、テスト概要、構成、条件、因子と水準、テスト項目、手順、合否判定基準、テスト結果、報告書を記載する。

変更により影響しそうな機能の「因子と水準」を明示し、レビューし易いように工夫した。

また、機能仕様書の個々の機能のテスト漏れを防止するため、総合テスト設計書では機能仕様書の章番号を明記することで、トレーサビリティを実現した。

③ 【テスト自動化】

テスト自動化支援ソフトウェアに「テストの実行手順」と、その「結果確認の手順」をまとめた「テストケース」を組み込んで自動化環境を構築した。ソフトウェア開発期間中にソフトウェアを変更して、社内でリリースするたびに、自動テストを実行することで、デグレードの発生を早期検出でき、早期対策が打てるようにした。

6.改善による変化や効果

【変化】

XDDP の適用当初、設計者の作業負荷が増えるため、開発部の組織を挙げてのトップダウンによる推進体制と、設計者の意見を取り入れながら XDDP に改善を加えることで、全ての派生開発で XDDP の適用を浸透させた。

その一方専任のテストエンジニアチームが設立されたことで、一部の設計者にバグの摘出をテストエンジニアチームに依存するような傾向が出てきた。

そこで、総合テスト工程への不具合流出を防止する目的で、単体、結合テスト工程と総合テスト工程の間にフェーズゲート「結合テスト完了確認」を設けた。単体結合テスト工程の品質が確保されたことを確認してから、総合テスト工程に着手するようにした。

従来、品質保証部が実施していたシステムテストは、機能を確認するテストであったが、テストエンジニアチームによる総合テストを強化することで、顧客のニーズを反映していることを確認するため、顧客の運用ライフサイクルに沿ったテストに変更した。

【効果】

XDDP 適用と、総合テスト強化、テスト自動化の 3 つの施策により、総合テスト工程からシステムテスト工程への不具合の流出率が半減した。

従来の機能仕様書は、変更後の仕様のみしか記載がなかったが、XDDP では変更前と変更後の仕様を「見える化」したので、より変更箇所が明確となった。また、従来の機能仕様書だけでなく、XDDP の 3 点セットの成果物に対するレビューも実施するようになったので、レビュー密度が高まり、仕様の誤り発見や漏れ防止ができるようになった。

一方、設計者は、総合テストを実施しないことで、作り込み工程と単体、結合テストの時間が確保でき、注力できるようになった。

7.改善活動の妥当性確認

【改善の内容および実現方法の妥当性】

今回のプロセス改善の施策の適用前と適用後で、後工程への不具合流出率を比較すると、適用後の方が、不具合流出率が大幅に低減したので、プロセス改善の施策が品質改善に貢献していると判断できる。

【残存課題】

市場への不具合流出をなくすためには、更なるプロセス改善が必要である。変更開発だけでなく、新規開発においても作り込み工程（上流工程）での仕様漏れをなくすこと、コーディングでの作り込み誤りをなくすこと、テスト工程で不具合を後工程に流出させないことが課題である。

A.参考文献

<タイトル>

サービス提供組織でのプロセス評価・改善の試み

<サブタイトル>

サービスのための CMMI 試行適用評価と他手法との比較

<発表者>

氏名(ふりがな): 大瀧 陽悦(おおたき ようえつ)

所属: NEC ソリューションイノベータ(株)

<共同執筆者>

氏名(ふりがな): 中野 典子(なかの のりこ)

所属: NEC ソリューションイノベータ(株)

氏名(ふりがな): 勝野 直樹(かつの なおき)

所属: NEC ソリューションイノベータ(株)

氏名(ふりがな): 岩崎 孝彦(いわさき たかひこ)

所属: NEC ソリューションイノベータ(株)

<要旨・主張したい点>

サービス提供部門では ITIL をベースに活動が進められていることが多いが、ITIL をどの程度活用しているか、外部からは知ることができない場合が多い

ITIL の提供状況をチェックするツールも整備されてきているが、CMMI ファミリーでもサービスのための CMMI が整備されている

サービス提供業務のプロセス評価・改善で ITIL ベースのチェックツールとサービスのための CMMI を併行に適用した結果の比較評価、適用結果をベースにサービスのための CMMI を活用していくに当たって改善した点、更にサービスのための CMMI を4部門に簡易適用した際の分析結果を示す

<キーワード>

サービスのための CMMI ITIL TIPA

<想定する聴衆>

サービス提供部門のプロセス改善メンバ

<活動時期>

2015/11～継続中

<活動状況>

- 着想の段階(アイデア・構想の発表)
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他(変更を実施し、一部結果は明確になった)

<発表内容>

1.背景

上位組織の状況としては、会社統合に伴いサービスを提供するグループが全国に存在することになったが、各グループ内にどのような作業があり、その作業をどのような手順で遂行しているのか組織全体として明確でなく、改善施策の立案が困難になっていた。

発表者の動機・目的としては、開発のための CMMI の入門研修インストラクタ、リードアプレイザとして、開発のための CMMI の社内研修や適用を進めてきた。サービスのための CMMI については、サービスというと社内では直ぐ ITIL が現場で連想され、中々適用推進までは持ち込めなかった。しかし、サービスのための CMMI は海外では適用が伸びており、社内で活用できない仕組みなのか、活用するためにはどのような整備が必要かを明確にしたかった。

2.改善したいこと

組織内に複数のグループがあり、それぞれサービス提供の業務を担当しているが、提供しているサービスの内容そのものも組織全体としては明らかでない。サービス提供組織として、ITIL の知識習得には取り組んできているが、ITIL をどれ程深く業務に生かしているか明確でなく、それぞれのグループの業務実施手順も明確でない。そこで、各グループでの提供サービスの内容、実施手順を明らかにし、良いグループの実施手順で全体を統一していきたい。

CMMI の適用を推進する立場からは、実際の現場でどのように活用できるか知見が内部に蓄積されておらず、また他手法との優位性も明確になっていなかった。そこで、CMMI を他手法と比較する共に、優位性及び有効性が確認できた場合には現場での利用方法を確立したい。

3.ここまで活動の経緯

3.1 初期適用

ITIL を使っている組織の評価手法の一つである TIPA による内部監査を実施する部門が組織内にあり、その内部監査に同席する形でサービスのための CMMI の簡易適用を進めることができた(CMMI 公式アプレイザル クラス C 相当)。

3.2 初期適用を通じた他手法との比較

初期適用を進めるに当たっては、CMMI の評価単位であるプラクティスレベルでの確認を準備したが、初期適用及び他の類似手法と比較するとプラクティスレベルでの確認では項目数が不足していることが明確になった。

4.これまでに得られたこと

4.1 サービスのための CMMI はより包括的に現状を評価

初期適用部門ではこれまで ITIL をベースに改善を進めてきており、それ以外の仕組みへの取り組みはあまりなかった。そこで、サービスのための CMMI を適用することで、プロジェクト管理の領域や組織的な改善活動の領域で改善候補が多くあがった。

4.2 確認項目はプラクティスからサブプラクティスに

初期適用ではプラクティスレベルで確認を進めたが、他手法との項目数の比較、及び記述内容も確認し、サブプラクティスレベルで確認を進めるように仕組みを変更した。この変更では次の点が期待される：

- ・他手法と確認項目数のレベルで遜色がない
- ・サービスに十分な経験がないメンバが確認活動に対応しても確認漏れが防止できる

5.組織の改善に向けて

初期適用を通じてサービスのための CMMI 活用の改善点が抽出されたので、改善したツールを活用し、組織内での試行適用を更に3部門で進めた。また、CMMI 1.3 版まではセキュリティの要件はないが、サービス提供部門ではセキュリティ・セーフティ確立の要望もあり、CMMI のチェック活動期間中に ISMS によるチェックも併せて実施した。当初、報告書の統合も目指したが、それぞれの活動のチェックメンバ間の日程調整が付かず、また ISMS のチェック項目も多かったため、今回報告書の統合は見送った。CMMI の Next Generation ではセーフティ・セキュリティも盛り込まれており、本活動と ISMS によるチェック活動の連携は引き続き検討していきたい。

6. 改善による変化や効果

6.1 見える化の進展

初期適用を通じて、対象組織の作業実態を明確にすことができた。評価に CMMI を使うことで、提供するサービスの構築やサービス提供のプロセス評価に留まらず、管理面も含めより包括的に活動の見える化が図られた。

6.2 横並び評価による組織的改善

複数部門の評価結果の横並びの分析により、組織として改善に取組むべき項目が明確にあがつた。

7. 改善活動の妥当性確認

7.1 サービスのための CMMI 活用レベルの適性さ

CMMI はプラクティスが評価で使われる単位だが、サービスのための CMMI はこのレベルでは他手法と比べ項目数が少なく、サブプラクティスを評価単位とした方が他手法の確認項目数に近くなる。サブプラクティスにも基本的な項目が記載されている場合が多い。

7.2 確認項目漏れの向上

確認項目をサブプラクティスとしたことで、プラクティスレベルの確認では漏れが発生するかもしれない項目も漏れなく確認可能となった。

7.3 サービスのための CMMI の評価結果が部門に有効であること

現在は個別業務での改善項目、組織として改善すべき項目まで明確になった。それらの改善に具体的に取組むに当たっては、費用対効果を事前に明確にして活動を進める予定である。

A. 参考文献

<タイトル>

ISO26262「確証方策」のより効率的な実施の取り組み

<サブタイトル>

～組織・プロジェクトの特性に応じたフレキシブルな実施を目指して～

<発表者>

氏名（ふりがな）：菅沼由美子（すがぬま ゆみこ）

所属：パナソニック株式会社 オートモーティブ＆インダストリアルシステムズ社

技術本部 プラットフォーム開発センター システム技術開発部 機能安全推進課

<共同執筆者>

氏名（ふりがな）：

所属：

<要旨・主張したい点>

ISO26262 の要求の中に、第三者による 3 種類の「確証方策」があり、成果物に対する ISO26262 要求の遵守評価や確証、安全プロセス遵守の評価、機能安全達成の評価などが要求されている。この確証方策を、3 種類それぞれ個別に、フルに実施することは、リソースの潤沢なプロジェクト、組織では問題ないが、小規模プロジェクト、組織では負担となる。また、プロジェクトの規模（開発期間、人員など）、関連する組織のリソース（技術職能、品質職能の工数とスキルなど）等により、確証方策の効率的な目的達成の実施形態が異なると考えた。そこで、組織やプロジェクトに応じた最適な方法で確証方策を実施するための、実施方法の選択肢を構築すべく、取り組みを行った。イベントの統合による効率化、視点の整理による効率化の、二つの効率化方針で取り組みを行い、4 つの選択肢を構築することにより、確証方策の実施の効率化を推進することが出来た。これらの取り組みと 4 つの選択肢を共有する。

<キーワード>

ISO26262 車載 機能安全 確証方策 機能安全監査 確証レビュー 機能安全アセスメント
システムチック故障 開発プロセス

<想定する聴衆>

車載機能安全の確証方策を担当される方、SEPG、SQA

<活動時期>

2010 年～（継続中）

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

近年の機能安全要求の高まりに伴い、車載開発では、顧客から要求を受けた場合、車載機能安全規格 ISO26262 に基づき、第三者による、下記の 3 種類の「確証方策」の実施を求められる。

- ・確証レビュー：作業成果物の ISO26262 の要求への準拠性の評価、確証
- ・機能安全監査：要求される開発プロセス（システム、ハード、ソフトの安全関連プロセス）の遵守の評価
- ・機能安全アセスメント：達成した機能安全の評価

これらの確証方策により、安全計画（安全関連の開発計画）の評価、システムチック故障の回避、ランダムハードウェア故障による安全目標侵害のリスクの評価、などを行い、機能安全の達成を総合的に評価する。確証レビュー、機能安全監査は機能安全アセスメントの入力として用いることが出来る。

また、筆者は、パナソニックの車載部門コーポレートの SQA、SEPG として、傘下の製品開発事業場での機能安全対応を推進するため、仕組み構築、人材育成、開発プロジェクトの機能安全支援などを行ってきた。本発表では、この中の、確証方策に関する取り組みについて、これまで検討してきた内容を共有する。

2.改善したいこと

確証方策を、3 種類それぞれ個別に、フルに実施することは、リソースの潤沢なプロジェクト、組織では問題ないが、小規模プロジェクト、組織では負担となる。また、プロジェクトの規模（開発期間、人員など）、関連する組織のリソース（技術職能、品質職能の工数とスキルなど）等により、確証方策の効率的な目的達成の実施形態が異なると考えた。

ここで、確証方策の目的は、機能安全の達成の評価…安全計画（安全関連の開発計画）の評価、システムチック故障の回避、ランダムハードウェア故障による安全目標侵害のリスクの評価、等を用いた、総合的な評価である。

そこで、確証方策の目的を効率的に達成すべく、下記の取り組みを開始した。

『組織やプロジェクトに応じ、最適な方法で確証方策を実施するための、実施方法の選択肢を構築する。』

3.改善策の概要

2 で述べた「改善したいこと」について、下記の取り組みを行った。

『組織やプロジェクトに応じ、最適な方法で確証方策を実施するための、実施方法の選択肢を構築する。』

(1)イベントの統合

…プロジェクトがかかわるイベントの増加を抑えるべく、イベントを統合する形で効率的な実施を目指す。

(2)視点の整理

…成果物ごとの確証方策の複数実施を抑えるべく、3 つの確証方策の確認視点や実施範囲を、成果物ベースで整理し、再定義する形で効率的な実施を目指す。

4.改善策の内容

3 で述べた(1)(2)の内容について述べる。

(1)イベントの統合

3 種類の確証方策を別々のイベントとして実施する場合には、プロジェクトごとに第三者確認が 3 イベント、別途 SQA 監査を行っている場合には 4 イベントになる。また確証レビュー、機能安全監査、SQA 監査は通常、それぞれ数回実施される。全確証方策をそのまま実施することは、大規模かつリソースが潤沢なプロジェクト及び組織では可能であるが、小規模プロジェクトには負担が大きくなる。そこで、小規模プロジェクトを想定して、イベント回数を減らし、統合イベントとして実施する実施方法を検討した。

取組①：SQA 監査と機能安全監査の統合

従来から弊社のソフト開発プロジェクトは、組織の開発プロセスを使用し、プロセスの遵守を第三者が確認していたが、2012

年より、開発プロセスの定義を、ハードウェア開発を含むシステム開発全体に拡張した。

これに伴い、SQA 監査も、ソフト開発プロセスに限らず、システム開発全体に対して実施可能とした。

SQA 監査の対象をシステム開発全体とすることで、機能安全監査（システム、ハード、ソフトの安全関連プロセス遵守確認）を SQA 監査とマージすることが可能になった。これにより、監査イベントは 1 種類のみに出来る。

取組②：確証レビューと機能安全監査の統合

確証レビューの対象は成果物、機能安全監査の対象はプロセスであり、その確認対象は異なる。しかし、弊社のプロセス監査チェックリストは、プロセスのアクティビティ毎に、そのアウトプットである成果物を通して、プロセス遵守を確認する構成となっていることから、成果物に対する ISO26262 の遵守確認を機能安全監査に統合できると考えた。すなわち、成果物の規格準拠確認を、プロセスの遵守確認と同時に実施する。このために、成果物ベースの確証レビューチェックリストを、プロセスのアクティビティごとに並べ替え、プロセス監査チェックリストとマージした。この際に分かったこととして、弊社プロセスにおいては、機能安全監査チェック項目の 63%が、確証レビューチェック項目を参照可能であった。このため、確証レビューと機能安全監査をマージすることで、効率的にこれらの確認方策を実施できるようになった。

実施の際は、機能安全監査の中で確証レビューを同時に行うことを想定しているため、監査部門（品質職能を想定）による確証レビューの実施（技術職能による実施を想定）が可能となる。この前提として、確認方策実施者のスキル管理が必要となる。人材育成と認定の仕組みとして、車載開発のプロセス監査員育成の仕組み、機能安全関係者の育成の仕組みを、それぞれ構築しており、機能安全監査と確認レビューの実施者は、これらの仕組みを活用して育成することが出来る。

(2) 視点の整理

確認方策は、最終的には、機能安全アセスメントで、機能安全の達成を確認する。確認レビューと機能安全監査は機能安全アセスメントの入力となる。そこで、機能安全アセスメントで機能安全の達成を評価できるための情報をダブりなく提供するというポイントで、確認方策を見直した。

取組③：視点ごとに確認イベントを整理

…成果物ごとに、確認レビューと機能安全監査の視点を、再度明確にして視点のダブりをなくす。

確認方策の視点を検討し、下記 3 つの視点でのイベントに再構築した

- ・プロセス & 規格準拠の確認
- ・次工程に対する確認
- ・機能安全の達成

各視点のチェックリストを構築し、現在、プロジェクトでの試行を行っている。

取組④：成果物ごとに視点を統合

…取り組み③で構築した 3 イベントを、視点に着目したまま、成果物ごとに統合する。

：

実施方法の選択肢をまとめる。

	選択肢
(1) イベントの統合	取組①：SQA 監査と機能安全監査を統合
	取組②：確認レビューと機能安全監査を統合
(2) 視点の整理	取組③：視点ごとに確認イベントを整理
	② 成果物ごとに視点を統合

5. 改善による変化や効果

これらの選択肢により、

弊社のすべての組織、プロジェクトにおいて、確認方策の効率的な実施を可能とすることができた。

選択肢の特徴をまとめる

選択肢	メリット	注意点
イベントの統合	①SQA 監査と機能安全監査を統合	<ul style="list-style-type: none"> ・イベント回数削減 (15→11) ・監査を統合 (SQA が機能安全監査を担当) できる
	②確認レビューと機能安全監査を統合	<ul style="list-style-type: none"> ・イベント回数削減(15→8) ・機能安全監査チェック項目の 63%で、確認レビューとの重複を削減 ・機能安全監査員が確認レビューを担当できる
視点の整理	③視点ごとに確認イベントを整理	<ul style="list-style-type: none"> ・視点が明確で確認が行いやすい (視点により担当者を適切に分けることが出来る) ・機能安全達成を強化 (次工程への確認により機能安全達成のリスクを低減) ・どのイベントで何を保証しているのかが、より明確 (機能安全アセスメントでの参照が容易)
	④成果物ごとに視点を統合	<ul style="list-style-type: none"> ・小規模開発では、③が更に効率化できる

A.参考文献

- ①SPI-Japan 2011 「プロセス監査のシステム製品への拡張～機能安全を追い風に～」 パナソニック(株) 菅沼由美子
- ②SPI-Japan 2012 「車載用機能安全規格 ISO26262 に対応した高信頼開発プロセス～機能安全開発を実現するソフト・ハードの協調～」 パナソニック(株) 安倍秀二
- ③SPI-Japan 2012 「車載ソフトウェア搭載製品の機能安全監査と審査」 パナソニック(株) 菅沼由美子
- ④SQiP2012 「システム、ハード、ソフト全プロセスに渡る システムプロセス監査の実施～車載機能安全規格要求への対応～」 パナソニック(株) 菅沼由美子
- ⑤SQiP2015 「「SQA監査」と「確認レビュー(ISO26262対応)」の融合～SQAの更なる役立ちに向けて～」 パンソニック(株) 菅沼由美子
- ⑥SPI-Japan 2015 「「SQuBOK」と「ワークショップ」を活用した SQA 育成～こんな SQA を育てよう！～」 パナソニック(株) 菅沼由美子
- ⑦SPI-Japan 2015 「機能安全開発を支える安全文化の確立と実践～機能安全開発スキル強化とスキル認定制度～」 パナソニック(株) 安倍秀二

3A1 「ふりかえりを用いたレビュー方法の改善」 木村慎吾(インテック)

<タイトル>

ふりかえりを用いたレビュー方法の改善

<サブタイトル>

<発表者>

氏名（ふりがな）：木村 慎吾（きむら しんご）

所属： インテック

<要旨・主張したい点>

開発経験のない初心者だらけのチームが製品開発の中で経験した、レビュー方法の改善についてお伝えする。チームメンバーが初心者だったため、機能（プログラムを含む）確認に関してはリーダである私が一人で実施していた。その属人化したレビューでは問題があったため、チームメンバーがボトムアップ的に改善を重ねてきた。具体的には「ふりかえり」というアジャイル開発の手法を使って改善を推進させた。その結果、メンバ間での機能レビューやプログラム作成時の暗黙知の共有など品質向上につながる改善がもたらされた。

<キーワード>

- ・アジャイル開発
- ・ふりかえり
- ・レビュー
- ・チケット管理
- ・Git
- ・PullRequest

<想定する聴衆>

- ・ソフトウェアエンジニア
- ・プロジェクトマネージャ

<活動時期>

2008年～現在

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

主体：自社パッケージの製品開発チーム（メンバは3名～10名程度）

- ・ただし、開発当初はリーダ以外に開発経験者がいないチーム

きっかけ：課題のボトムアップ提案

- ・開発プロセスにはアジャイル開発を採用
- ・アジャイル開発のプラクティスである「ふりかえり」によって改善提案が発生

2.改善したいこと

・作成した機能の品質を向上させるために、レビュー方法の改善を考えた
(状況)

- ・開発当初、開発経験者はリーダのみであったため、レビューは一人で実施していた
(改善のねらい)
- ・メンバそれぞれがレビューできるようなることで、全体として品質を向上させたい。
- ・製品の改修作業が続くので、保守性を上げることで全体の品質を向上させたい。

3.改善策を導き出した経緯 ※1回目（今回紹介する改善は2段階ある）

メンバは開発経験がないため、他者の成果物をレビューする能力や余裕がなく開発当初はリーダが一人でレビューを実施した。成長させるために、まずは知識の取得を積極的に行っていった。具体的には読書会など勉強会を開催した。また、ふりかえりで問題点を明確にすることでメンバの成長を促していった。

その結果、メンバが成長したため成果物の作成速度が速くなってきた。そのころになると1つの成果物にかけられるレビュー時間が短くなりレビュー内容の粒度が当初より荒くなっていた。そのような状況はよくない状況だとメンバが認識し、ふりかえりで問題としてあがつた。

4.改善策の内容

ふりかえりでメンバが提案してきた改善内容としては、レビューはメンバが持ち回りで実施しようということであった。

5.改善策の実現方法

具体的には、終了した作業については朝会で他のメンバにレビューを依頼するという方針で進めることになった。

※2回目

3.改善策を導き出した経緯

メンバ間レビューを運用していく中で、プログラムのレビューについての問題がふりかえりであがつた。問題としてはバグではなく書き方のお作法的な部分の指摘がやりにくいということであった。この原因の1つとして、プログラムがすでにメインブランチに取り込まれており、わざわざ再修正してもらうには手間がかかるので言いにくいということがあった。また、開発期間も長くなっていることもあり、暗黙知も多くなっていることも影響を与えていた。

このような状態を放置していくと、今後のメンテナンス作業がどんどん修正しにくくなることが想定された。そのため、今後を考えるとこれらに対して対策をする必要があると考えた。

4.改善策の内容

メインブランチに取り込まれる前にプログラムを指摘しやすい方法がないか検討した。その方法として PullRequest 方式というものがあるとわかった。そこで、その方式を採用し、追加・改修したプログラムをメインブランチに取り込む前に必ずレビューすることにした。

5.改善策の実現方法

PullRequest という方式を提供しているツールを採用し、運用することにした。ただし、これはソース管理のリポジトリを Git というものにしなければならず、製品自体は Subversion で管理していたため、既存の製品に関連しない新機能の開発の時にこれを用いることにした。

6.改善による変化や効果

メンバ間でレビューすることで属人化を排除することができた。これによって、開発当初は知っている個所が担当個所だけであったものが、ほかの個所に対しても知識をつけることができた。

これによって、追加・修正時に関連している個所を早い段階で気づけるようになった。そのため、同じような問題の取りこぼしなどが削減でき、品質向上として効果が出たと考えている。

さらに、プログラムに対するレビューの改善もできた。長期間、開発を実施しているため、暗黙知がとても多くなっていました。そのため暗黙知も PullRequest でのチェックによって、以前より伝達できるようになった。特に後から入った若いメンバの教育的な意味としてはとても効果的が出たと考えている。

上記の改善により、当初考えていた、チーム全体としての品質向上と保守性の向上による品質向上については一定の成果が出たと考えている。

7.改善活動の妥当性確認

ふりかえりを用いることで、チームが課題と感じていた問題を解決できることはよかったです。一気にすべてを改善していくのではなく、そのときにもっとも必要な個所からできるので、負担なく確実に実施できる点がよかったです。また、ボトムアップ的な取り組みなため、メンバそれぞれが課題として認識しているので、改善に対する取り組みが積極的であり、効果も出やすいと感じた。

最後に、残課題して新機能開発のみが PullRequest 方式に対応しているため、既存の製品開発には対応できていないことがある。意識があがり、既存の改修時にもプログラムの細かい指摘をするというモチベーションにはなったが、仕組みとして存在していないので、その時の状況に左右されやすいという課題もでできている。

改善で意識をあげることは重要だが、それを維持していく仕組みを作りこんでいくことの重要性にも気付いた。そのため、今後はすべて PullRequest 方式に移行していくことを考えている。

- 以上 -

A.参考文献

<タイトル>

やらされ感の壁を越えて、現場と一緒に改善を考えるための第一歩

<サブタイトル>

プロセス監査役から、改善推進役へのシフトチェンジ

<発表者>

氏名（ふりがな）：江崎 美保（えざき みほ）

所属：株式会社 日新システムズ 未来戦略室

<共同執筆者>

氏名（ふりがな）：陸野 礼子（りくの れいこ）

所属：株式会社 日新システムズ 未来戦略室

氏名（ふりがな）：前川 直也（まえかわ なおや）

所属：株式会社 日新システムズ 未来戦略室

<要旨・主張したい点>

品質保証部門が主体になって、標準プロセスの定義、プロジェクト管理に関する社内基準の策定、その運用状況を監査する仕組みなど品質保証体系を構築し、この10年で一定の効果は出てきた。一方、現場は、要求品質や開発モデルなど変化があるなかで、今までと同じように品質を確保し、開発プロセスも遵守しなければならないと考える品質保証部門との間に見えない壁が出来つつあった。この問題を解決すべくはじめた取り組みについて報告する。

<キーワード>

プロセス改善、やらされ感、ふりかえり、コミュニケーション

<想定する聴衆>

※品質保証部門の方、改善推進者（特にこれから改善を開始する、または開始したばかりの方々）など

<活動時期>

開始時期：2015年10月。現在継続中

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

弊社では、受託ソフトウェアの品質向上を目的として、2004年から品質保証部門が、標準プロセスの定義、プロジェクト管理に関する社内基準の策定、プロジェクトの定量評価のための品質指標の設定などを行い、その運用状況を監査し、社内基準の定着状況やQCDの目標達成、クレーム費用の削減など成果を確認しつつ、検出した課題に対して改善を進めてきた。

一方、現場においては、開発モデルの変化やユーザの求める品質の多様化、今まで以上に低コスト・短納期を迫られることで、品質を確保しつつ開発プロセスも遵守しつつ開発効率をUPすることが必要となってきた。

2.改善したいこと

品質保証部門の主な取り組みは、以下の通りである。

- 標準プロセスの定義、各種基準類の制定・改訂・維持といった品質保証体系の構築・保守
- 課題の早期検出と対策状況の追跡、品質指標を用いた定量的評価の実施による失敗プロジェクトを未然に防ぐプロセス監査
- 品質教育やマネジメント教育の実施

上記の結果として、目標とした納期やコストを大幅に超過するような失敗プロジェクトや納品後障害の減少など、一定の効果は得てきた。

自分自身、実際にプロセス監査担当として、基準通りに開発プロセスが実施されているか監査し、抽出したプロジェクト課題の解決状況を確認し、期限までに解決しない場合は上級管理者へのエスカレーションを実施してきた。失敗プロジェクトなど問題が発生した場合は、原因分析し、再発防止として標準プロセスや基準の見直しも行ってきた。

しかしながら、上記の活動を進める中で、現場からは「今の基準は、現状のプロジェクトの進め方に合っていない」といった声を、この2、3年よく耳にするようになった。現場の開発プロセスと品質保証のやり方にギャップが出来始め、次第に品質保証部門からの「やらされ仕事」という意識が強くなり、品質保証部門との間に「見えない壁」が出来始めた。

そこで、出来てしまった壁を取り除けるように、そして現場がやらされ感で課題解決するのではなく、自分たちで課題解決していくようなマインドづくりを目標に改善に取り組み始めた。

3.改善策を導き出した経緯

まず、現場の考え方を知るために、現場のプロジェクトリーダを対象に現状の品質保証活動に関するアンケートを実施した。

KPTを参考に、Keep（良かったので継続していきたいこと）」「Problem（問題点、改善したいこと）」「Try（改善提案、やってみたいこと）」で意見を収集し、整理した。

その結果、以下のような課題を抽出した。

- 品質保証の各活動、基準・標準の策定目的を、現場は理解していない
- 現場の進め方と品質保証活動の方法があつてない
- プロジェクト実績が、組織的に共有できていない

1点目は、現場は、制定した基準・標準をプロジェクトで運用する上で「何のための基準なのか」「何のためのプロセス監査なのか」の目的を理解できていなかつたので、作業を実施する上での指示されたルールでしかなく、記載された作業をすることが目的となっていました。

また、基準・標準の制定・改訂など仕組みの見直しは、失敗プロジェクトを契機とすることが多かつた。失敗原因を分析し、再発防止のために見直ししていたのだが、特に問題なくプロジェクトを運営していた部門やプロジェクトリーダにとっては、見直しされた理由を十分に説明されないまま運用されるケースもあった。当然、現場では自ら決めた・考へたという意識を持つことができず、「ただやることが増えた」と感じ、それも壁を作った原因となっていたと考える。

2点目は、現場の開発の進め方が変わってきてているにも関わらず、品質保証部門のプロジェクトへのかかわり方がプロセス監査立ち上げ当時と同じままであることが原因だった。「現状のプロジェクトの進め方に合っていない」という現場からの声は聴いていたものの、例えば実際のプロジェクトの進捗会議やレビューなどに積極的に参加して現場の状況を知ろうとすることも少なかったため、現場は形式的に作業するようになり、品質保証活動の形骸化を促進することになったと考える。

3点目は、現場においても、プロジェクトでうまくできることや失敗したことをフィードバックする習慣がなく、プロジェクトの進め方の良い点・悪い点の共有ができていない状況であった。品質保証だけでなくプロジェクト運営全般で、これまでのツールや帳票などを、現場視点で使いやすい、効果のあるものに変えようといった自律的な改善も行われることなく、「自分たちに合わない…」と思いつつルールに合わせて実施することで、やらされ感をさらに強くしていたのではないかと考える。

4.改善策の内容

プロセス監査の方法や既存の基準・標準を見直しするには、現場の仕事のやり方を知ることからはじめないと、また現場とギャップが生じてしまうため、まずは現状を把握するために、以下の2つを実施した。

■プロジェクトの朝会への参加

今までのプロセス監査の中で、現場と顔を合わせて実施する監査レビューは、プロジェクトの開発計画レビュー、試験計画レビューと、出荷判定の3回のみであったが、毎日開催される朝会にも参加し、プロジェクトメンバと話をすることで、進捗状況やプロジェクト課題を確認するようにした。

■プロジェクトふりかえりの実施

先に述べたとおり、現場においても、プロジェクトメンバや関連メンバでプロジェクトの結果を議論し、情報共有する機会もない。品質保証部門も全プロジェクトの朝会に参加するほどリソースもない。プロジェクトがどのように運営され、どんな問題点をもっているのか、どんな工夫をしているかを把握したいのは、品質保証部門も現場も同じと考え、プロジェクトのふりかえりを実施することとした。

5.改善策の実現方法

■プロジェクトの朝会への参加

自分が品質保証を担当するプロジェクトは、アジャイルを活用し始めたプロジェクトであり、毎朝15~30分間の朝会を実施していたため、そこに参加するようにした。

朝会の中で報告される、当日の作業予定、課題解決状況などを確認しつつ、プロジェクトリーダのプロジェクト管理の方法、使用ツールや管理帳票の使用状況なども合わせて確認した。また、品質保証部門としては、アジャイル開発の品質保証の方法を模索していたため、現場がどのように品質をレビューやテストを実施し、品質をコントロールしようとしているのかも合わせて調査した。今まで実施してきたプロセス監査や定量評価は継続できるのか、それ以外に品質の見える化はできるのか、プロジェクトリーダにも協力してもらい、品質状況も確認していった。

■プロジェクトふりかえりの実施

弊社では、SQAとしての品質保証部門はあるが、SEPGの役割担当は不在で改善活動を進めていたため、どうしても現場が受け身になってしまう傾向になり、一時的な改善により効果を出すことはできても、組織戦略的なプロセス改善を推進するための取組みが定着しにくいという弱点もあった。そのため、弊社では、2015年度下期に、自律的なプロセス改善、現場コンサルティング、人材育成などを戦略的に推進していく未来戦略室が新設され、各事業部にSEPGを配置し、自律改善を加速できるような体制が構築された。そこで、品質保証部門と未来戦略室とSEPGと連携して、プロジェクトふりかえりの推進をはじめた。

（1）全部門への目的の説明

開始するにあたって、今までの反省から、まずはふりかえり実施の目的・目標を理解し、納得してスタートできるよう、全

部門に説明会を実施した。ふりかえりの目的、対象プロジェクトの定義、実施方法などを、部門ごとに行い、実施に対する意見、疑問、不安などの現場の声を聞くことを一番重要視した。

その結果、「管理職が入ると、本音が出なくて建前で終わってしまうのでは…」「報告会しても、結局叱られて終わるような…」と現場の懸念点が分かったので、『誰もが発言でき、お互いに聴きあえる』ような空気づくりと、最後に『ほめられる』を引き出せる場になるように、ファシリテーション役として実際のふりかえりの場に参加した。初めは、どうしても管理職から突っ込みが入り、場が静まりかえることもあったため、ディスカッションの進行も苦労したが、ふりかえり終了後に管理職にふりかえりの主旨を再度説明しにいくなどフォローし、懸念点を少しずつ解消していく努力を実施した。

また、当初は、全プロジェクトを対象とし実施する計画であったが、プロジェクトの規模も進め方チーム構成も様々なので、現場をよく知る各事業部の SEPG にふりかえり対象の選定と日程調整を実施してもらい、全員が最低 1 回はふりかえりに参加できるように工夫した。

（2）ふりかえり結果の共有

部門内で報告会を開催し、プロジェクト毎のふりかえりの結果を共有できるようにした。プロジェクト完了時に、都度部門のメンバを全員集めるのは難しかったので、毎月の部門会議などの機会を利用し、報告会を開催したり、各事業部の SEPG も相互で報告会に参加して事業部間の情報共有もできるようにするなど、工夫して実施している。

6. 改善による変化や効果

プロジェクトの朝会への参加やプロジェクトのふりかえりを実施していく中で、以下のような気づき、効果を得ることができた。

- 現場は、思った以上に混沌としており、品質保証部門から指摘されたことを対処しようにも「何をどうしたらいいのか」を考えている余裕がない、また考えるのが苦手な人もいる。一緒に原因と改善方法を考える人が必要である。これについては、現状 SEPG が支援できるので、少しずつ改善されていくと考える。
- プロセス監査による指摘は、現場にとっては優先度の低く、別に優先度の高い課題を抱えている場合がある。これらは、現場の状況を把握できていない場合に発生するので、今後も継続して朝会など身近なところから入り込んで課題を共有することが大切である。
- ふりかえりの効果として、チーム力・コミュニケーションの大切さに現場が気づき始めた。
「自分の担当部分以外の、技術的な知識やスキルを身に着けることができた」「みんなで課題を共有できた」「今まで個人バラバラで他人が何をしているか知らないことも多かったが、作業の見える化ができた」など、もっとコミュニケーションを取ろうという意識が芽生えてきた。
- ふりかえりするにも、課題分析力が必要
問題の深堀できていないので、せっかく自分たちで考えた改善策が実行に移しにくい内容になっている。今回は、ふりかえりが実施されることを最優先にしたので、ふりかえりの内容については深く言及していない。今後は、ふりかえり自体のふりかえりを実施し、より効果的にノウハウや教訓が次のプロジェクトに活かせられるようなディスカッションへ成長させる。

7. 改善活動の妥当性確認

まだ妥当性確認には至っていない。現時点では、現場と品質保証部門の間の見えない壁は残っているが、朝会などで現場に入りこむことによって現場と品質保証担当者間では都度ディスカッションができている。現在は、未来戦略室に所属することとなり役割は変わってしまったが、今後は監査役ではなく改善推進役として、ふりかえりの参加などは継続し、品質保証活動に対する現場の課題を抽出・解決を支援することで、見えない壁を解消していく。

また、今後は定量的にもふりかえり実施の効果を測定、評価し、現場の自律改善を促進していきたい。

A. 参考文献

<タイトル>

「対話」を加えた KPT ふりかえりによる、チームの活性化

<サブタイトル>

見えなかつたもの見える化し、行動を変える！！

<発表者>

氏名（ふりがな）：松浦豪一（まつうらひでかず）

所属：（株）富士通マーケティング GLOVIA 事業本部 サービス・サポート統括部 サービスピジネス部

<共同執筆者>

氏名（ふりがな）：長岡桃子（ながおかももこ）

所属：富士通株式会社

グローバルビジネス戦略本部 サービスプラットフォーム戦略企画室 インキュベーションセンター

<要旨・主張したい点>

KPTを使ったふりかえりを本来の目的である健全な開発プロセスの学習のために利用する。

「対話」しながら振り返りをすることで、チームの自己組織化を促す。

<キーワード>

KPT, ふりかえり、自己組織化, TPS, KAIZEN 活動

<想定する聴衆>

・自律改善活動、アジャイル開発に興味のある方

・SEPG

・SQA

<活動時期>

2016 年から本格的にアジャイルの教育に取り入れている。

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

.背景

ふりかえりの手法である KPT は、アジャイル開発/ウォーターフォール開発問わず多くのソフトウェア開発の現場で利用されている。しかし、ふりかえりが形骸化してしまい、思うような結果を出ないということが多く起こっている。具体的にはマンネリ化して活発に意見が出ない、多くの問題が発生していっこうに改善されない、同類の問題がたびたび発生するなどである。

2.改善したいこと

KPT を使ったふりかえりが機能しなくなるのは、チーム内の「対話」ができていないことが原因であると仮定した。KPT を使ったふりかえりを行う際に、「対話」できるようにすることで、チームを活性化させる。KPT でふりかえりをする上で、「対話」できるようにすることで、チームに合った健全なやり方を見つけるツールとしての効果が得られるようにする。なお、今回の事例ではアジャイル開発を実践していることを前提としている。

3.改善策を導き出した経緯

KAIZEN 活動を推進した経験やアジャイル支援をした時にチームのメンバーが対話できていないことが多かった。問題が根深いのは、本人達に対話ができていないという自覚がないことである。富士通では、リーダークラスに対して、ファシリテーションの教育を実施しているため、リーダーはファシリテーションの概要を理解している。ただし、実践のレベルには大きく個人差がある。実践に慣れていないリーダーがファシリテーションを実施した場合の問題点は以下のとおりである。

問題 1) オープンクエスチョンができない。

「○○だったんじゃないの？」

問題 2) 詰問をしてしまう。

「なぜ、○○しないんだ！！」

問題 3) 否定をしてしまう。

メンバの意見に対して

「○○ダメだろう」と否定してしまう。

以上の問題は、リーダー自身がコマンドコントロール型のリーダーシップに慣れていることに起因すると考えられる。一方チームメンバーとしては以下の問題がある。

問題 4) すべての意見が表出されない。

リーダー主導で発言者が決まることが多く、その場合、発言の機会が与えられるのは一部のメンバーとなる。

他にあるか？と聞かれても、自分の意見に自信が持てなかつたり、否定されるのが嫌と感じたりしていいにくい。

問題 5) 自分の意見を持とうとしない。

自分は、意見してはいけない立場だと勝手に思い込んでしまう（例：協力会社の社員）。

問題 6) 他人の意見の問題点をさがす。

自分の意見は言わないで、人の意見を否定する。

改良案をだけを言う。

問題 7) 他人の意見に「同じです」しか言わない。

考えていないのをごまかすために同じですという。

同じですと言っておけば自分に意見をもとめられない。

上記のように振舞うことが当たり前になっており、チームメンバーはリーダーの指示に従うだけで、自分の意見を言わないようになってしまう。チーム活動を改善するような意見が出てこないため、形骸化、マンネリ化、問題未解決の状態になる。

自己組織化されたチームづくりには、多様な意見交換を行う、開発チームの「対話」が必要である。開発チームが「対話」することによって、自己組織化され、改善することによってチームが健全なやり方を見つけることができる。「対話」には、コミュニケーション

を通じてお互いが自らのフレームワークを問い合わせていくプロセスであるという定義もある。⁽³⁾ 筆者はチーム活動の改善に直接寄与する KPT に着目した。KPT を実施する時に対話が促進されれば、チームは自己組織化すると考えた。

あわせて活動すると自然に対話するようになるようなシンプルな KPT グラウンドルールを検討する、という取り組みを進めてきた。富士通が提供しているワークショップ型のアジャイル実践教育で指導し、受講者の職場で実践してもらった内容について紹介する。

4.改善策の内容

KPT を使ったふりかえりに KAIZEN 活動のアプローチを加える。

- ・付箋紙を使った KPT ボード (KPT の整理・整頓)
- ・問題対私たちを意識したフォーメーション (半円型)
- ・リーダ以外のチーム運営 (リーダレス)

※アジャイル開発における KPT を使ったふりかえりについては、以下の参考資料に書かれている内容が含まれている。

- ・これだけ！KPT あらゆるプロセスを成果につなげる最強のカイゼンフレームワーク
- ・プロジェクトファシリテーション 値値と原則編

5.改善策の実現方法

アジャイル開発の教育で実際に実施することで、行動や経験を得させるようにしている。

教育内容はそのまま各自の職場に持ち帰って実践できる手順になるよう設計している。

以下の手順で実施する。

- 0) グラウンドルールおよび心構え
 - ① 積極的に話すこと
 - ・当事者意識を持つこと
 - ・議題に集中すること
 - ② 一人で話しそぎないこと。人の発言をさえぎることは禁止
 - ・話していない人にも想いあり
 - ③ 問題に向かってはなすこと
 - ・人に向かって話すのではなく、問題に向かって話し、問題対私たちの構図を作る
- 1) イテレーションの出来事を開始日から終了日までふりかえる (30分)
- 2) 前回の TRY/Problem の整理 (必要・不要を判断して不要を棄てる) をする。
実践できた TRY→KEEP 解決した Problem→はずす
- 3) 新しい KEEP と Problem を洗い出す。 (5分)
KEEP は 1 人 1 件以上
- 4) KPT ボードの前に半円上にならび、KEEP からひとりひとつずつ順番に出していく。 (5分)
全員付箋とペンをもつ。半円状態をキープする。腕組みはしない。
…距離的には肩と肩がふれあうかあわないかギリギリくらい。
終わったら次というテンポでポンポンしていく。2秒感覚…司会の指示は最初だけ
同じ課題に対して別のアプローチでも自分の意見を言い、そのまま貼る
- 5) Problem について、エピソードを確認する。(10分)
「いつ」「どんなふうに」「どう思った」を確認する。

問題発生時のエピソードを共有することで、チームの行動の理由やパターンが共有される。行動の変更を検討し、これが後の意識の変化につながる。

なお、この時の司会は持ち回りとする。どのメンバーが司会となっても同じ流れであることが重要である。平等感が醸成される。

- 6) TRY を洗い出す。…TRY は具体的な行動を書く。（5分）

だれがやっても同じになるように。問題解決策でなくとも、問題検出を早めるための施策でも良い。

他人が出した Problem や Keep に対して TRY を出すようにする。

- 7) KPT ボードの前に半円上にならび、TRY を張り出していく。（5分）

- 8) 実施する項目を決める。（3つくらい）（5分）

※付箋紙は3色用意し、毎回ちがう色を利用する。同じ色になったものはすべて捨てる。

どうしても継続的に検討したければ改めて出すようにする。

ふりかりの KEEP、Problem、TRY をたくさん貼ることにはあまり意味がない。今のチームの状態を見る化することが目的のひとつであるため、最新の KPT が貼り出されてあれば十分である。また、Kaizen 活動の観点からも、整理整頓された状態を保つことを意識させている。

必要であれば記録として個人所有のノートに貼るといったことをすればよい。

施策の理由：

1. 付箋紙を利用する理由

→誰が何枚の意見を保持しているか、全体で何枚書けたか、手がとまつたかを見極めできる。

…見えないものを見る化

→補記、追記、新規起票がすばやくできる。…スピード感をもつ

→いざれはがれる。（はがれるまで放置されるようなら捨ててよい。）…整理整頓

→付箋紙にサインペンで書くと、書かれた内容が記憶に残りやすい。

2. 半円状にならぶ理由

→ミーティングに批判的、懐疑的であるか体勢でわかつてしまう。…見えないものを見る化

防御型 腕を組む

反論型 あごに手をあてる

降参型 後ろで腕を組む

後退型 チームの和から離れてしまう

対面型 だれかとだけ話す

演説型 一人が一方的に話してしまう

評論型 誰かの意見にコメントしようとしている

→他人がすぐ近くにいるため、体勢の差に自分で気付きやすい。

→早く終わらせたいと思う。…自働化（無意識にはたらきかける）

3. テンポをあげる理由

→集中させる。…制約を与えて考えさせる

→テンポにあわせて出していくと、批判的な態度が取れなくなる。…自働化（無意識にはたらきかける）

→テンポにあわせてだそうとして積極的になる。…行動の変更による意識の変化

6.改善による変化や効果

筆者による指導（アジャイル開発の教育および、現場支援）を経験したチームでは、自己組織化されて活動ができており、その成長について SPI Japan でもいくつかのチームが発表をしている。

教育では、KPT を使ったふりかえりを2日間で3回行うが、以下の変化がおきる。

- ・チームメンバーの心理的距離が縮まる。
- ・チームメンバー全員が自然に自分のカードを出す。
- ・時間内に会議を終わらせることができる。
- ・対話によって、複合的な案をつくり選択することができる。

KAIZEN 活動とアジャイル開発を比較した場合、アジャイル開発のほうが継続しやすい。

→アジャイル開発は、開発プロセスと管理プロセスが同期しており、改善がすぐに反映されるため、

効果を実践しやすい。

以下の問題が発生していない。

- 思うような結果を出すことができない。
- 活発な意見が出ない、
- 問題が多く発生する。
- 同類の問題がたびたび発生する。

KWS⁽⁵⁾では問題を深堀し、組織として横断的な施策づくりをすることを促している。KPT に対話をできるようにしただけでは横断的な施策が課題として残る。この課題に対しては以下の施策のどちらかで対応している。

横断施策 1：競争

施策を横展開するのではなく、同じ初期施策を行い、複数のチームの結果を共有させる。競争意識を芽生えさせると、改善スピードが速まる。

横断施策 2：分割

アジャイル開発実施中に人数とチーム数を増やしたい場合は、経験を積んだ既存メンバーを複数チームに分割し、それぞれに新規メンバーを追加することで、経験知を伝達する。

7.改善活動の妥当性確認

ふりかえりが定期的に実施され、改善が継続的に行われている。

小集団活動の特徴もあるが、「対話」ができるようになると、同じ考え方で作業できる。

共有している内容に誤りがある場合も、誤りのまま共有されるというリスクもある。合意を得やすいため、反対意見がでなくなる。結果的にチームの健全性が損なわれる可能性がある。

定期的にメンバを入れ替えたり、要求をだす側が適切な NG を出すことが必要に応じて求められる。

以上のような改善を、筆者は、チームを見守りながら、健全であるかを

確認して適切な課題を与えるながら伝道師型の指導で行ってきた。無意識に実践している改善を意識させて実践することでチームのさらなる改善を促す役割もある。指導者とチームとの「対話」も成功の鍵であると考えている。

A.参考文献

- (1)これだけ! KPT, 天野 勝, ISBN978-4-7991-0275-6
- (2)リーソフトウェア開発と組織改革, Mary and Tom Poppendieck,
訳:依田智夫, 依田 光江, ISBN978-4-04-868741-6
- (3)チーム脳の作り方, 清宮 普美代, ISBN978-4-87290-405-5
- (4)プロジェクトファシリテーション 値値と原則編, (株)永和システムマネジメント 平鍋 健児, 天野 勝
- (5)「KPT」と「なぜなぜ分析」を応用した KWS 振り返りの研究,

SQIP 第一分科会 グループ B ソルトウェアプロセス評価・カイゼン (27 年度 2011 年)

3B1 「品質コックピット構築による品質改善」 松井伸一(NEC ソリューションイノベータ)

<タイトル>

品質コックピット構築による品質改善

<サブタイトル>

定量的データに基づいた品質アラーム表示からの組織的プロセス改善

<発表者>

氏名 (ふりがな) : 松井 伸一 (まつい しんいち)

所属 : NEC ソリューションイノベータ(株) 技術統括本部 品質保証室

<共同執筆者>

氏名 (ふりがな) :

所属 :

<要旨・主張したい点>

全社組織的プロセス改善活動の一環として、プロセスツール「品質コックピット」を構築し、運用した。

「品質コックピット」を活用することで、プロセス遵守に役立つとともに定量的データに基づいたアラーム情報を発信し、QCD + CS の評価状況が一目でわかる。この仕組みにより、プロジェクトの品質状況を俯瞰して認識でき、問題プロジェクトに対して早期解決を図ることができる。

CMMI レベル 5 に相当する活動である。

<キーワード>

品質改善、定量的データ、QCD、CS、計測、プロセス品質、プロダクト品質、CS品質、プロセスツール、組織的プロセス改善

<想定する聴衆>

SEPG、SQA

<活動時期>

2007 年～現在

<活動状況>

- 着想の段階（アイデア・構想の発表）…統合会社として
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階 …統合前の NEC システムテクノロジー社として
- その他 ()

<発表内容>

1.背景

NEC ソリューションイノベータ社は 2014 年に NEC グループの国内ソフトウェア会社 7 社が統合した 12,000 人規模の会社である。北海道から沖縄まで全国に分散し、NEC グループが展開する社会ソリューション事業における中核会社として、システムインテグレーション事業、サービス事業、基盤ソフトウェア開発事業を展開している。

2014 年度は準備期間とし、2015 年度から統合会社としての QMS 活動を開始し、国内最大級の ISO9001 認証を取得している。しかしながら、プロセスに準拠していても、一部において品質問題を抑えきれないこともある。さらなる品質を向上するためにも統合会社として積極的に改善活動の強化に取り組んでいるところである。

ここでは、統合前の 1 社、NEC システムテクノロジー社にて取り組んだ品質改善活動を事例として紹介し、今後の取り組みを考察したい。

2.改善したいこと

統合前の 1 社、NEC システムテクノロジー社では、組織的に QMS プロセスを遵守しており、QMS 活動の維持はできていた。しかしながら、品質の確保、向上につながらないことがあった。その原因として、QMS 活動として年度単位で品質情報を集計しており、ソフトウェア品質改善サイクルの間隔として年間では長めであったことが、品質問題の発生を抑制するには至らなかったと考えられる。このことから、改善すべき点は、品質問題発生の予兆を早期に検知し、予防できるプロセスの構築であるとした。

3.改善策を導き出した経緯

ソフトウェアの品質を測る指標として「プロセス品質」「プロダクト品質」「C S 品質」がある。それぞれの定量的な実績値は事業部門にて管理されており、QMS 活動として年度単位で内部監査や外部監査で整理された品質情報を共有されていた。しかしながら、SQA が定期的に定量的な品質状況を把握するには手間と時間がかかるため、整理された情報をタイミングよく参照できない状況であった。それは、人手を介した集計であり、事業部門毎に多段階となっていることでオーバーヘッドが大きかったことによる。そのため、QMS に準拠しているものの、SQA は、プロジェクトの品質状況やリスクをタイミングよく検知できず、組織として品質問題の発生を抑制するための予防策を取ることができづらい状況であった。

そこで、組織的に品質問題の発生を抑制するプロセスに改善し、そのプロセスを遵守するためのプロセスツールの導入を考えた。

4.改善策の内容

品質保証の仕組みとして、遂行中のプロジェクト全体を「Q C D + C S」の観点で俯瞰できるよう、定量的データを収集し、全社のW e b ページに視覚的に表示して公開することとした。このW e b ページを「品質コックピット」と呼び、月次単位で品質状況を一覧で見えるようにすることを目指した。その際、経営者にも品質状況をアラームとして一目で確認できるよう、青黄赤の信号表示を点灯することとした。

「品質コックピット」というプロセスツールを活用しながら、事業部門に対して教育やトレーニングを繰り返すことでも運用を定着させ、組織的なプロセス改善の取り組みが図れ、定量的な効果につなぐことができた。

5.改善策の実現方法

「プロセス品質」として QMS にて規定している出荷前の「出荷判定」に加え、遂行中プロジェクトのリスク度合いを確認する「リスク診断」を定量的に測定できるよう指標を定めた。「出荷判定」の場合は、出荷判定の合格数、出荷判定日遵守数から納期達成率を導出して基準を設定した。その基準を達成していれば青信号、未達に近づけば黄信号、未達であれば赤信号を点灯させた。「リスク診断」の場合は、リスク度合いを測る質問に答えていくだけで自動的にリスク度合いを判定するように点数化した。さらに、月次でリスク診断するので、危険度レベルに変化がなくても質問回答で前回との差分があれば、リスク検知することも考慮した。

プロジェクト遂行中は、活動コストも収集し、内部失敗コストが発生しないか、受注金額をベースとした事業計画に対し、予算

に近づけば黄信号、超過すれば赤信号を点灯することとした。

「プロダクト品質」として「出荷後バグ」情報を取り込んだ。出荷後に発生したバグの発生件数および重要度をランク別に収集した。QMS にて計画した目標値に対し、目標値に近づけば黄信号、超過すれば赤信号を点灯することとした。

「CS品質」として出荷後、四半期毎にお客様に対するアンケート調査を実施し、5段階評価の結果を集計し、QMS にて計画した目標値に対し、目標値に近づけば黄信号、超過すれば赤信号を点灯することとした。

このような定量的計測値を「品質コックピット」というプロセスツールでアラーム表示させることで、次の品質問題の対策に取り組めるようなプロセスへ改善した。

6. 改善による変化や効果

「品質コックピット」は、全社公開のWebページといえども、参照権限を部門長以上に設定し、他プロジェクトへの不必要的公開を避けるように配慮している。これにより、プロジェクト側は、プロジェクト報告の一環として SQA へ状況を報告するのみで、公開先やエスカレーションを考えることなく、気兼ねなく発信することができるようになった。

「品質コックピット」の仕組みは、SQA をはじめ、事業部門の部門長以上は「品質コックピット」を参照できるようにし、全会議でも状況を確認することでアラーム発生に気づき易くなり、早期段階で対策に取り組むことができるようになった。

組織的にプロセス改善を図る中で、この仕組みを活用することで、次のような定量的な効果があった。

品質面では、出荷後バグの発生率を1/2にまで、コスト面では、リスク検知率を上昇させることにより、失敗PJの発生率を1/3にまで下げる事ができた。納期面では、出荷納期達成率を95%以上をキープすることができた。CS面では、3以下の低評価の割合を下げる事ができた。

プロジェクトの品質状況の監視ができ、プロジェクト側も品質問題を積み残すことがなくなり、プロジェクトの遂行が大幅に改善された。SQAとしては、経営者に対する品質状況の報告が、定量的に、かつタイムリーに行えるようになり、業務効率の向上が図れることになった。また、継続的に組織としてのプロセス改善も図れるようになった。

これは、CMMI レベル5に相当する活動になったといえる。

7. 改善活動の妥当性確認

NECシステムテクノロジー社では、ISO9001の認証維持とともに、プロセス品質だけでなく、プロダクト品質やCS品質も確保、向上につなぐしくみを構築することができた。それは、プロジェクト個別の活動ではなく、NECシステムテクノロジー社全体でトップマネジメントもからめて組織的に取り組んだことでガバナンスも効き、プロジェクト側へ受け入れられる体制ができたことによる。

統合会社では大幅に規模が拡大し、今まで通りの仕組みをそのまま適応するのは難しくなっている。今後は、規模が大幅に拡大した統合会社でも、さらに強化する形での組織的なプロジェクトマネジメントやリアルタイムマネジメント等の施策を進めていき、その中で従来から積み重ねたノウハウを取り込んだ「品質コックピット」を再構築することを構想している。現時点では構想段階であるが、「品質コックピット」を活用した全社組織的プロセス改善の仕組みが統合会社に定着できれば、より品質向上につながるものと確信している。

A. 参考文献

なし

3B2 「社会インフラシステムにおけるプロジェクトの異常予測手法の適用」 飯村拓志(東芝)

<タイトル>

社会インフラシステムにおけるプロジェクトの異常予測手法の適用

<サブタイトル>

眠ったデータ資産を活用したらできちゃいました！ プロジェクトの異常予測！

<発表者>

氏名（ふりがな）： 飯村 拓志（いいむら たかゆき）

所属：東芝 インフラシステムソリューション社 府中事業所 技術・情報システム部

<共同執筆者>

氏名（ふりがな）： 森 俊樹（もり としき）

所属：東芝 インダストリアル ICT ソリューション社 IoT テクノロジーセンター プロセス・品質技術開発部

氏名（ふりがな）： 平井 桂子

所属：東芝 エネルギーシステムソリューション社 府中事業所 電力系統システム部

氏名（ふりがな）： 片川 毅哲

所属：東芝 エネルギーシステムソリューション社 府中事業所 電力系統システム部

氏名（ふりがな）： 野村 奈央

所属：東芝 エネルギーシステムソリューション社 府中事業所 電力系統システム部

氏名（ふりがな）： 斎藤 秀充

所属：東芝 エネルギーシステムソリューション社 府中事業所 電力系統システム部

氏名（ふりがな）： 乙丸 裕輝

所属：東芝 エネルギーシステムソリューション社 府中事業所 電力系統システム部

<要旨・主張したい点>

適用部門は、元々、豊富なデータ資産をもっていたが、従来の統計的品質管理手法は、サンプル数やデータ品質などの制約が大きく、十分に活用できなかった。ナイーブベイズによるプロジェクト異常予測は、データのばらつきや欠損に強く、かつ、ベテランプロジェクトリーダの経験に基づく「このまま進むと危ない」という感覚を自然に定量化できるため、比較的スムーズに適用できた。また、検討の過程で、今後の改善活動に向けたプロジェクト関係者との客観的データに基づく有意義な議論も実施できた。

<キーワード>

データ活用、異常予測、過去のプロジェクトデータ資産、ナイーブベイズ

<想定する聴衆>

システム/ソフトウェア開発のプロジェクト計画や管理に携わる関係者

<活動時期>

2015年10月から開始し、2016年5月時点で継続中

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

当社の電力系統監視制御システムを開発している部門では、発電所の調整制御を行うシステム、変電所・需要家への電力供給・自然エネルギーの安定供給を図る監視制御システムなどを提供している。当該部門の開発において、電力インフラに関する製品の性質上、大規模で長期にわたるプロジェクトが多い。そのため、プロジェクトの異常検出が後工程になるほど、影響範囲が大きく、対応が難しくなるといった課題がある。また、プロジェクト管理自体が人依存となっており、管理のバラつきや定性的管理も課題となっている。

2.改善したいこと

プロジェクト管理について、下記の課題を改善したい。

- ・プロジェクトの異常検出の遅延
- ・プロジェクト管理の人依存

3.改善策を導き出した経緯

本事例で紹介する部門では、CMMI の簡易アプレイザルを定期的に実施している。簡易アプレイザルの結果、プロジェクトの管理データは蓄積されているが、そのデータを十分に活用できていないといった弱みが出ていた。そこで、蓄積した管理データを活用して組織やプロジェクトの異常や品質傾向を見える化する方針となつた。

4.改善策の内容

当社では、「ナイーブベイズ識別器を用いたプロジェクト失敗リスク予測モデル」が研究されている。本手法を用いてプロジェクトの異常を予測するモデルを開発し、モデルの活用方法をプロセスに組込む改善策とした。

(本事例では、プロジェクトの失敗ではなく、異常が発生するリスクを予測した)

5.改善策の実現方法

当社で研究されている「プロジェクトの異常予測手法」では、プロジェクトの管理データはもちろんだが、プロジェクトの異常・正常の閾値が必要となる。その上で、管理データを使って異常を予測するモデルを構築する流れとなる。本事例での改善策実現までの流れを下記に記載する。

- ① 本事例の部門では、異常・正常の閾値が整備されていなかったため、まずは、定性的観点でプロジェクトの異常・正常を識別するメトリクスを「Y：目的変数」として定義
- ② ①で定義した「Y」に影響する可能性のあるメトリクス「X：説明変数」を決定し、「X」のデータを収集
- ③ 蓄積されているプロジェクトの管理データから、サンプリングした約 100 件のデータを使用し、定量的観点で「X」と「Y」に関する関係があるか確認
- ④ ナイーブベイズ識別器による、初期予測モデルを構築
- ⑤ AIC(Akaike's Information Criterion)の指標を参考に初期モデルを改良
- ⑥ プロジェクトの異常予測モデルの試行

6.改善による変化や効果

- 効果の観点（定性的）

- ・プロジェクトの早い段階から、異常が発生する確率を予測できるので、早期に異常を検出しやすくなつた。
- ・異常が発生する度合いを定量的に確認できるため、関係者間でリスクを共有しやすくなつた。（図 1 参照）

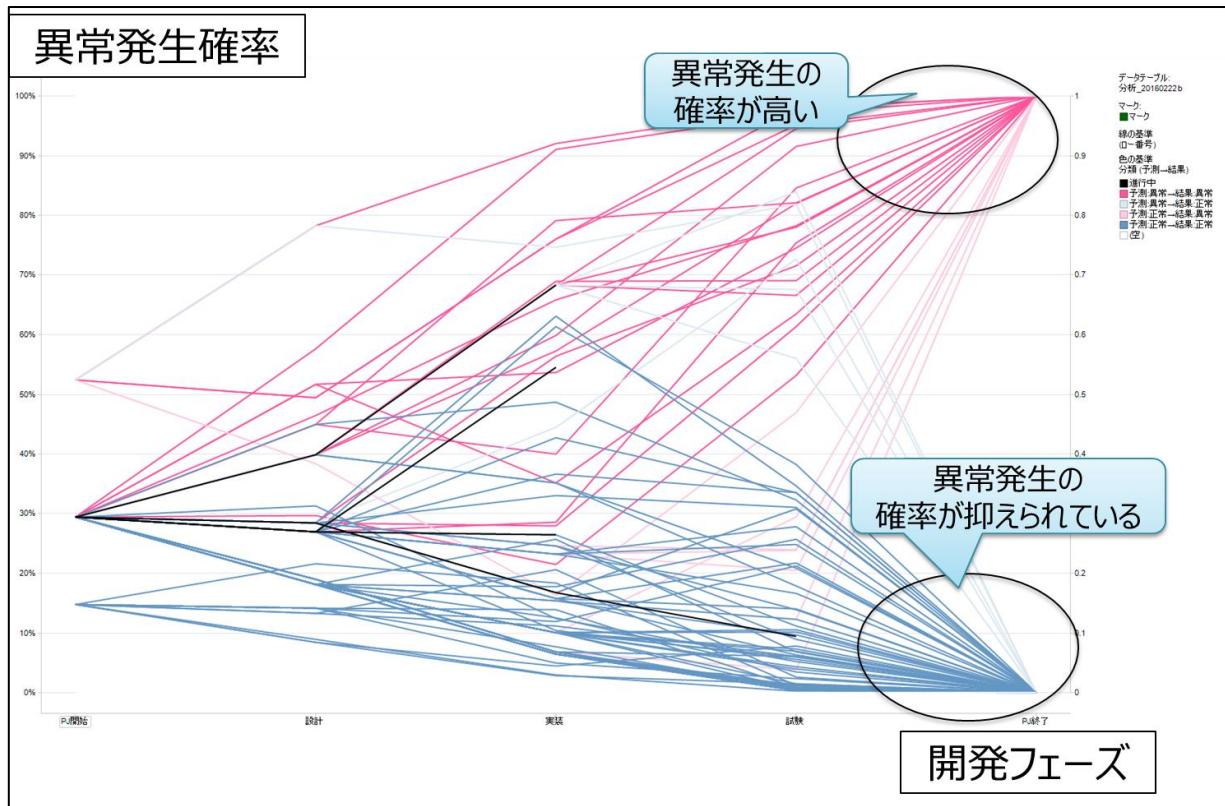


図 16 プロジェクトの異常予測の結果

- 変化（新しい課題の発生）

今まででは、異常が発生し、異常の内容が明確になっている段階で対策を打つため、対策が出しやすい部分もあった。しかし、異常予測では、異常が発生する前に予測されるため、具体的な対策を練りこいといった課題も出ている。
(モデル構築時に定義した「X」の粒度が粗く、異常をコントロールするレベルまで落とし込めていない)

7.改善活動の妥当性確認

構築したプロジェクトの異常予測モデルに、進行中のプロジェクトの管理データを入力し、予測した結果について、プロジェクトリーダが得ている状況感覚と合っているか確認した。

A.参考文献

- ・森 俊樹, 覚井 真吾, 田村 朱麗, “プロジェクト失敗リスク予測モデル”, 東芝レビューVol69 No.1, 2014
- ・森 俊樹, 覚井 真吾, 田村 朱麗, 藤巻 昇, “プロジェクト失敗リスク予測モデルの構築”, プロジェクトマネジメント学会誌, Vol.15, No.4, 2013, p.3-8.
- ・T. Mori, S. Tamura, and S. Kakui, “Incremental Estimation of Project Failure Risk with Naïve Bayes Classifier,” In Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Baltimore, MD, USA, 2013-10, ACM/IEEE, P.283-286.

3B3 「品質特性に基づく品質メトリクスの定義と活用」相澤武(インテック)

<タイトル>

品質特性に基づく品質メトリクスの定義と活用

<サブタイトル>

なし

<発表者>

氏名（ふりがな）：相澤 武（あいざわ たけし）

所属：株式会社インテック 生産本部 品質保証部

<共同執筆者>

なし

<要旨・主張したい点>

一般的に品質特性に基づく品質メトリクスを使用し品質評価を行うといふと、敷居が高い印象を持たれるケースが多い。

しかし、今回の品質メトリクス定義の過程で行った社内調査の結果を見ると、新たに定義する品質メトリクスのいくつかは、既に品質評価で使用されていることが確認でき、決して敷居は高くないということがわかった。

本発表では、品質特性に基づき定義した品質メトリクスと、それをプロジェクトの中で活用するために行つた方策、特にプロジェクトメンバーがいかに敷居が高いという印象を持たずに品質メトリクスを活用できる仕組みにしたか、について紹介する。

<キーワード>

SQuaRE、ISO/IEC 25010、品質特性、品質メトリクス、品質計画書、品質目標

<想定する聴衆>

品質保証の担当者

<活動時期>

2015年3月～継続中

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

近年のIT業界の動向をみると、ITを活用した製品やサービスの種類や社会における役割が増えるにつれ、利用者が求める品質も、従来の必要な機能が正しく動作すること、といったものから、機能の正確性だけにとどまらず、利用目的や利用場面に応じて、安全性やセキュリティはもとより、快適さや楽しさ、また、ビジネスへの高度の貢献といったように多様化してきており、品質を取り巻く環境が大きく変化してきている。

社内においても、昨今ではお客さまから提示されるRFPには、品質特性に基づく要求事項が明示されていることが一般的となつてきている。また、Webアプリケーションにおける脆弱性対策のように、RFP等で明示されていなくても、きちんと対策を取ることが事業者側の義務となっているものもあり、これらに的確に応えられることが求められている。また、的確に応えるためには、お客さまとの間では共通の評価基準を持っておく必要がある。

これらの状況から、新たな品質に対する考え方として、品質特性を理解したうえ、お客さまとの共通の評価基準を持ち要求達成に向け、開発工程の中でどのように品質を担保していくかを計画し、実行することが、これまで以上にプロジェクトにおいては必要不可欠となつてきている。

2.改善したこと

当社においては、これまで業務プロセス標準の中でプロジェクト計画の一部として品質計画書作成が定義されていたが、前述した新たな品質に対する考え方を取り組むには、次のような課題があった。

- ・品質特性（旧6特性）別に品質目標は設定していたが、お客さま要求事項との関連が明示されていなく、要求事項として提示されている項目も、品質目標として明示的に設定されていない場合もあった。
- ・これまでの品質計画書では、品質目標、品質保証活動（レビューやテスト方針）、工程完了判定基準などを記載するフォーマットであったが、品質目標と品質保証活動や工程完了判定基準との間で相互参照性がなく、品質保証活動等が必ずしも品質目標達成のための活動内容と結びついていない面があった。
- ・社内において品質メトリクスの共通定義がなく、事業部門毎に独自の定義を使用していた。

上述した課題を解決するためには、以下の改善が必要であると考えた。

(1) 社内の品質メトリクス定義の共通化

お客さまとの間での共通の評価基準を持つにあたっては、社内での品質メトリクス定義の共通化が必要である。ただし、共通化にあたっては、事業部門毎に異なる業務特性を持っているという点を考慮する必要があった。

(2) 客観的かつ精度の高い品質評価・判断を可能にする

定義した品質メトリクスをもとに設定した品質目標と目標達成のための活動内容が関連付けられている必要がある。

3.改善策を導き出した経緯

社内の品質メトリクス定義の共通化を図るにあたっては、世の中にあるベストプラクティスをそのまま適用するのではなく、社内でも既に利用されている品質メトリクスを、利用頻度の高いものを中心に整理・集約し品質メトリクスとして定義した。 整理・集約は以下の手順で行った。また、この過程の中で、出荷判定時に使用している評価基準などがどのように設定されているのかを確認した。

(1) 社内事例の調査

社内事例は、次の2ケースのどちらかに該当する部門のプロジェクトを複数件ピックアップし調査した。

①既に出荷判定時のチェックリストで品質メトリクスが明確になっているケース

出荷判定時のチェックリストから品質メトリクスを調査

②チェックリスト化はされていないが、出荷判定時に何等かの品質メトリクスを用いて評価しているケース

下記にあげたドキュメントからプロジェクトで使用している品質メトリクスを調査

品質計画、テスト計画書・結果報告書、判定会議資料、プロジェクト完了報告書、品質保証データ実績値

(2) METI 品質メトリクスセットへのマッピング

調査した品質メトリクスを ISO/IEC 25010 (SQuaRE) の品質特性に基づき品質メトリクスが網羅的に整理されている「METI 品質メトリクスセット^(*)1)」へのマッピングを行った。

マッピングした結果を見ると、既に代替メトリクスなどを用いて品質評価を実施しているプロジェクトもあることがわかった。

(3) 標準品質メトリクスの定義

マッピングした結果をもとにメトリクスの取捨選択を行い「標準品質メトリクス」を定義した。

なお、取捨選択の際には METI 品質メトリクスセットで定義されているメトリクスの意味を損なわない範囲で、メトリクス名称がわかりにくいものについての名称変更やメトリクス値としての収集が当社においては妥当ではないものについては、チェックリストでの確認へ変更するなど適宜カスタマイズを行っている。

(*)1) METI 品質メトリクスセットについて

経済産業省ソフトウェアメトリクス高度化プロジェクトにおいて、国内に存在する品質メトリクスを利用実績をもとに集約し、「情報システム／ソフトウェアの品質メトリクスセット」として 2011 年度にとりまとめられたものである。

全 173 個のメトリクスが利用時の品質特性、製品の品質特性ごとに整理されている。

4.改善策の内容

改善策として次の 2 点を行った。

4-1.社内の品質メトリクス定義の共通化

品質メトリクスとして、計 50 個の品質メトリクスを定義した。品質特性・副特性レベルで、全てをカバーしている。

品質特性・品質副特性と品質メトリクスの関係は、図 1 で示す通りである。



図 1.システム／ソフトウェア製品の品質モデル

4-2.客観的かつ精度の高い品質評価・判断を可能にする

定義した品質メトリクスを使用して、客観的かつ精度の高い品質評価・判断を可能にできるように、品質計画書の改定を行つ

た。今回の改定で新たに作成したシートは次の 2 シートである。

(1) 要求事項一覧シート

「要求事項一覧」とは、お客様の業務や構築するシステムやサービスの特性に応じて求められるひとつひとつの要求事項を、品質特性・品質副特性に対応付けしたものである。特徴として次の 2 点がある。

①要求事項を一覧形式で俯瞰できる

要求事項と品質特性・品質副特性との対応付けを一覧形式で俯瞰できることにより、品質特性の観点から抜け漏れがないかを確認することができる。

②「要求事項一覧」と「品質目標と工程別活動内容」の相互参照が可能

「要求事項一覧」の要求事項欄と後述する「品質目標と工程別活動内容」のテーリング理由欄で、相互に参照することができる。これにより「要求事項一覧」に挙がった要求事項を、確実に「品質目標と工程別活動内容」に展開することができる。

(2) 品質目標と工程別活動内容シート

「品質目標と工程別活動内容」とは、プロジェクトの品質目標及び目標達成に向けた工程毎の活動内容を定義したものである。定義にあたっては、社内事例の調査で収集した情報を参考にした。特徴として次の 3 点がある。

①品質要求を品質特性で整理・仕様化

品質要求を品質特性・品質副特性の観点で整理することができる。

②品質メトリクスを用いた評価基準の設定

品質メトリクスの基準値は、客観的に判断できるように数値を原則としているが、数値化が難しいものは、できるだけ主観的な判断が排除できる項目を設定し、「YES/NO」で判定できるようにしている。

③開発の主要なマイルストンでの工程評価や出荷判定の判断に利用

品質目標達成に向けて、工程毎ではどのような活動を行えばよいかを計画時に決めておくことで、各工程の移行判定などで、確認すべき項目として利用することができる。

5.改善策の実現方法

改善策を全社に定着させるために、標準化とトレーニングの 2 つの施策を行った。

5-1.標準化

全社における品質の見える化のレベルを底上げし、客観的かつ精度の高い品質評価・判断を可能にすることを目的に「品質見える化ガイド」と「品質計画書サンプル」の作成と公開を行った。

(1) 品質見える化ガイド

このガイドの具体的な利用目的は次の通りである。

- ・品質見える化のための活動手順を明確にする
- ・品質計画の重要性と作成・利用方法の理解を促す
- ・社内の品質メトリクス定義の共通化を図る
- ・品質メトリクスの工程内での評価方法の理解を促す

(2) 品質計画書サンプル

品質計画書サンプルの中では、今回定義した品質メトリクスを、「全社汎用」パターンとして位置付け、「品質目標と工程別活動内容」などのサンプルを掲載している。今後、事業部門毎の業務特性に応じたパターンを順次整備していく予定である。（業務特性とは例えば、SI、パッケージ、金融・公共などの業種などのこと）

5-2.トレーニング

e ラーニング教材の開発・公開と個別支援を行った。

(1) e ラーニング教材

開発したeラーニング教材は次の3コースである。いずれもコースも受講対象者は、品質計画作成に携わるプロジェクトマネージャ及び品質計画レビューや品質計画作成支援に携わる品質担当者を想定している。

- ①品質見える化ガイドの概要（目的は、品質見える化ガイドの開発背景や概要について理解すること）
- ②品質特性に基づく品質計画の作成（目的は、品質計画の作成手順を理解すること）
- ③RFPで学ぶ品質特性（目的は、品質特性の理解を深めること）

（2）個別支援

個別支援では、eラーニング受講者が実プロジェクトでの品質計画作成に関わる質疑応答や課題対応などを行う。

6.改善による変化や効果

今回の取り組みを行ったことで、現時点では以下にあげるような変化が出ている。

6-1.社内の品質メトリクス定義の共通化

標準品質メトリクスを定義したこと、例えば「レビュー指摘密度」など、これまで算出方法が部門により異なるものもあったが、標準の定義に合わせる動きも始めており、社内における品質メトリクス定義の共通化は徐々にできてきている。ただ、事業部門ごとに特性があるので、品質メトリクスによっては、いくつかのパターンが必要となると想定している。

6-2.客観的かつ精度の高い品質評価・判断を可能にする

品質計画書に新たに作成した「品質目標と工程別活動内容」シートを作成することで、工程移行や出荷判定時の評価基準が明確になった。

7.改善活動の妥当性確認

今回紹介した取り組みは、まだプロジェクトに適用を始めたばかりであり、適用しているプロジェクトの大部分が仕掛中であり、最終的にお客さまの多様化する要求を満たすことができたかどうかについては、今時点では、まだ結果が出ていない。

しかし、今回の取り組みによって、

- ・お客さまからの要求事項を品質特性の観点で整理する
- ・品質目標は立案するだけでなく、目標達成のための活動内容も合わせて考えることで、目標達成に向けた活動を確実に行うなどの点を意識してプロジェクトの遂行ができるようになり、これを継続していくことで、多様化するお客さまの要求事項にも応えられるのではないかと考える。

今後の課題としては、適用したプロジェクトでの適用結果の評価や事業部門毎の業務特性に応じたパターンの整備等が必要であると認識している。

A.参考文献

- 1.独立行政法人情報処理推進機構（IPA）技術本部 ソフトウェア高信頼化センター（SEC）、
つながる世界のソフトウェア品質ガイド、独立行政法人情報処理推進機構（IPA）、2015.05
- 2.SQuBOK 策定部会（編集）、ソフトウェア品質知識体系ガイド -SQuBOK Guide-（第2版）、2014.11
- 3.経済産業省ソフトウェアメトリクス高度化プロジェクトプロダクト品質メトリクス WG、
システム／ソフトウェア製品の品質要求定義と品質評価のためのメトリクスに関する調査報告書、2011.03

以上

3C1 「仮想化技術を使って、みんなでチャレンジ！」 佐藤直哉(東芝テック)

<タイトル>

仮想化技術を使って、みんなでチャレンジ！

<サブタイトル>

チャレンジャーを増やすための、仮想開発環境の構築

<発表者>

氏名（ふりがな）：佐藤 直哉（さとう なおや）

所属： 東芝テック株式会社

<共同執筆者>

氏名（ふりがな）： 末廣 龍夫（すえひろ たつお）

所属： 東芝テック株式会社

<共同執筆者>

氏名（ふりがな）： 平原 嘉幸（ひらはら よしゆき）

所属： 東芝テック株式会社

<要旨・主張したい点>

シミュレータを使ったソフトウェア開発を広めるためには、シミュレータ導入の敷居を下げる必要があった。vSphere 上の仮想マシンにシミュレータをインストールしたものを複数用意し、社内 LAN からリモートデスクトップで常時利用可能にした。使い方の教育も実施し、導入の敷居を下げたことで利用者が増え、様々な目的で活用されるようになった。

<キーワード>

仮想化技術、シミュレータ、シンクライアント、複合機、開発環境

<想定する聴衆>

ソフトウェアエンジニア、品質保証の担当者、ツール導入者

<活動時期>

2009 年～継続中

<活動状況>

- 着想の段階（アイデア・構想の発表）
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

東芝テックでは、全社ソフト開発力強化活動の一環として、2009年から実機レス情報交換会を発足し、SPI Japan 2013 にてこの活動が紹介された。当時、製品へ組み込むソフトウェア開発のために多くの実機が必要だったが、社長からこれらの実機を減らすような指示があったことが活動の動機。

2.改善したこと

トップダウンでソフトウェア開発用に作る実機の台数を削減して、シミュレータを使ったソフトウェア開発を進めることになった。シミュレータでやりたい要件はいくつか挙がってきたが、シミュレータの導入は敷居が高く、使ったことがない人にとってはどこまで活用できるのかといった不安があった。一方で、要件定義、仕様決め、製品のマニュアル作成、製品教育に活用できるのではないかといった期待もあった。

シミュレータ導入の敷居を下げて、気軽にシミュレータを使えるようにする。仮想化技術を使ったソフトウェア開発を普及させ、仮想化技術なしでは難しかったことに挑戦できるようにする。

3.改善策を導き出した経緯

シミュレータは筆者を含めて一部の技術者によって使われていたが、インストールが複雑だった。制限が多く、ハイスペックなPCが必要で、操作手順書なども未熟だったため、一般のソフトウェア技術者には嫌厭されていた。便利さをわかってもらえば、活用の幅は広いはずと考え、シミュレータの導入を簡単化する方法を考えた。開発用の複合機(実機)の予約表は常に予約でいっぱいだったため、本当はもっと使いたいが我慢している開発者も多いのではないかとの見込みもあった。

4.改善策の内容

シミュレータを不具合修正し、安定化。vSphere 上の仮想マシンにシミュレータをインストールし、必要台数を用意。更に、社内 LAN 経由でシンクライアントのような軽い端末からリモートデスクトップで常時利用可能にした。自動ビルト、自動デプロイ、自動更新の仕組みを入れ、最新の開発環境がいつでも使えるようにした。

5.改善策の実現方法

仮想マシンに Linux をインストールし、複数台コピーしてもコストがかからないようにした。Windows アプリのシミュレータを実行できるように、オープンソースソフトウェア(OSS)の Wine を活用した。リモートデスクトップ環境も OSS の XRDp を活用した。操作手順書を纏め、教育も実施し、予約の仕組みも用意した。



図 1: 実現方法

6.改善による変化や効果

予約表から利用状況が把握できるようになり、ユーザ数の増加が確認できた。活動が広まり、次の製品でも活用したい、なくなつたら困るという声が増え、自動評価をやりたいなど新たなニーズも得られた。実機だと、実機を壊してしまうかもしれないという不安から試せなかつたことも、仮想環境では安心して試せるというコメントも増えた。次の製品の要件定義にも活用されている。

2014B		2015A							2015B				
Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	
1.00	0.98	0.89	1.27	2.31	2.26	2.30	4.90	4.90	4.90	4.90	3.27	2.30	

表 1: 複合機開発でのシミュレータの利用状況 2014 年 2 月の利用状況（人・日）を 1 としたときの比率



図 2: ある日のシミュレータの予約状況

7.改善活動の妥当性確認

社内 LAN からのリモートデスクトップで常時最新のシミュレータ環境を利用できるようにしたことで、シミュレータ導入の敷居が下がり、利用が増えた。完璧なシミュレータでなければ、これ以上台数を減らすことは難しいのではないかという行き詰まり感があつたが、現状のシミュレータでも様々な活用ができることが分かった。削減できた実機のコスト以外にも、開発効率が上がった、実機ではできなかつたことができるという現場の声が大きな効果だと思う。シミュレータのユーザビリティを下げないように、今後も活動を継続していきたい。

A.参考文献

3C2 「あるある診断ツール」による保守/運用課題の見える化 室谷 隆(TIS)

<タイトル>

「あるある診断ツール」による保守/運用課題の見える化

<サブタイトル>

～さあはじめるぞ、改善の「か」…ka・si・ka～

<発表者>

氏名（ふりがな）：室谷 隆（むろや たかし）

所属：T I S 株式会社 生産革新本部 生産革新部

<共同執筆者>

氏名（ふりがな）：

所属：

<要旨・主張したい点>

「あるある診断ツール」は

- ・だれでも簡単に（約30分）
- ・現在のPJ状態をチェックするだけで

改善課題を見える化できる、エクセルで作られたツールである。

社内の保守PJや、運用PJに適用した結果、以下の効果が分かり、有用であることを確認した。

- ・PJの課題が明確になり、改善策が立案可能
- ・改善の前後比較で改善効果を見える化
- ・PJメンバーの認識相違が明確になり、コミュニケーションが活性化
- ・顧客との認識相違が明確になり、コミュニケーションが活性化し
 - －顧客と合意すべき改善ポイントが明確化
 - －顧客満足度の向上効果
- ・排除、削減、低減しなければならない課題の認識 ⇒ 教育効果
- ・同一部門内でのPJ比較により、改善策事例の取得

<キーワード>

保守、運用、生産性向上、品質向上、セルフアセスメント、改善、見える化、簡易ツール

<想定する聴衆>

保守、運用の生産性向上、品質向上などの改善に携わっている方、関心をもたれている方

PJマネージャ、PJリーダ、改善担当、品質担当

<活動時期>

2014/10～現在 繼続中

<活動状況>

- 着想の段階（アイデア・構想の発表）

- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他（ ）

<発表内容>

1.背景

昨今のシステム開発は、売上・利益ともに、保守開発の比率が高くなっている。保守の生産性向上/品質向上が喫緊の課題となっている。また保守と並行で運用を行なっている組織もあり、運用へのプロセス改善のニーズも高い。

2.改善したいこと

新規開発に関しては社内標準プロセスがあり、PPQA、工程開始/終了クライテリアなどの制度により生産性向上や品質向上の問題/課題への取組はできあがっている。しかしながら、保守/運用に関しては標準プロセスが未整備であり、日常業務で問題/課題への意識がしづらい状態となっている。

このため、保守/運用 PJ の生産性や品質に関する問題/課題を見る化し、改善に繋がる仕組みが必要である。

3.改善策を導き出した経緯

製造業で良く使われる「生産の 4M」にヒントを得て、保守のアウトプットの QCD を左右する要素として、「保守の 4 つの要因系」を想定した。

- ・人（スキル）
- ・顧客特性（業種・業態）
- ・システム特性（ハード/ソフト）
- ・管理（Management）

取り掛かりとして、管理（Management）系の課題を見る化するツールを以下のコンセプトで開発した。

- ・誰でも簡単に
- ・時間をかけずに（約 30 分）
- ・生産性や品質の低下を招いている、身近で発生している問題事象をチェックするだけで
- ・PJ の課題を、8 つの視点から見える化することができる
- ・オフラインでも使えるエクセルツール
- ・セルフアセスメントのための簡易ツール

4.改善策の内容

本ツールはセルフアセスメントツールの一種である。世の中に存在するセルフアセスメントツールは、アセスメントモデルのプラクティスが実施できているかを直接問うものが多く、プラクティスを理解していないと回答が難しく、時間がかかるものであった。このため、誰でも簡単に回答できるようにするために、設問は現在起こっている問題事象をチェックする方式とした。

また、課題の 8 つの視点であるが、保守開発は新規開発とは違い、修正箇所は少ないが、その影響調査とテストに工数がかかると言った特徴をもっている。（ふたごぶラクダと呼ばれている特徴）この特徴を踏まえ、以下の 8 つの視点で保守を捉えた。

1. 依頼/受付
2. 要件定義
3. 影響調査
4. 設計
5. コーディング/単体テスト
6. 品質確認テスト
7. リリース
8. 運用

設問への回答は「有る」、「どちらかと言えば有る」、「どちらかと言えば無い」、「無い」の 4 択としている。回答結果は 8 角形のレーダーチャートで示され、問題が存在する箇所が視覚的に分かるようになっている。比較は 8 つの視点だけでなく、設問項目単位でも可能となっている。また、回答結果データはエクスポート/インポートすることが可能なため、他者や、他 PJ、他部署などのデータと比較することができ、改善ポイントの見える化だけでなく、様々な使い方ができるようになっている。

5.改善策の実現方法

設問は、各視点で約 30 項目、全体で約 240 項目ある。各項目は共通フレーム 2013 のアクティビティや、SPEAK-IPA のプラクティス等を参考にしつつ、実施していなかった場合、起こるであろう問題を想定し作成した。

その文言は、なるべく短く、誰でもが理解できる用語を使うよう考慮した。

社内の正式リリースに当たり、いくつかの PJ に試用してもらった結果がフィードバックされ、正式な設問とした。

試用に当たっては、想定外の使われ方（顧客と一緒に実施）もあったが、良い知見をもたらすことができた。

6.改善による変化や効果

以下の効果が分かり、有用であることを確認した。また、使用現場から、運用業務を診断するコンテンツを作成して欲しいとの要望があり、「運用あるある診断～業務運用編～」を作成し、正式リリースした。

- ・PJ の課題が明確になり、改善策が立案可能
- ・改善の前後比較で改善効果見える化
- ・PJ メンバーの認識相違が明確になり、コミュニケーションが活性化
- ・顧客との認識相違が明確になり、コミュニケーションが活性化し
 - －顧客と合意すべき改善ポイントが明確化
 - －顧客満足度の向上効果
- ・排除、削減、低減しなければならない課題の認識 ⇒ 教育効果
- ・同一部門内での PJ 比較により、改善策事例の取得

7.改善活動の妥当性確認

当初の目的である保守開発の課題を見える化し、改善の PDCA を回す取り掛かりのツールとしての位置付けは十分満足し、利用者の増加、適用範囲、適用分野を拡大中である。現在も、改善施策の中心ツールとして機能の向上を図っている。

A. 参考文献：

- ・独立行政法人 情報処理推進機構 技術本部 ソフトウェア・エンジニアリング・センター編、プロセス改善ナビゲーションガイド～自律改善編～、2013
- ・独立行政法人 情報処理推進機構 技術本部 ソフトウェア・エンジニアリング・センター編、共通フレーム 2013、2013
- ・独立行政法人 情報処理推進機構 技術本部 ソフトウェア・エンジニアリング・センター、SPEAK-IPA (Rev.1.0.2.0)

<タイトル>

信頼度成長曲線の導入による統合テストの改善

<サブタイトル>

<発表者>

氏名 (ふりがな) : 中村 伸裕 (なかむら のぶひろ)

所属 : 住友電工情報システム株式会社 Q C D 改善推進部

<共同執筆者>

氏名 (ふりがな) :

所属 :

<要旨・主張したい点>

従来、統合テスト工程での品質管理は、表計算ソフトで検出欠陥数の累積値をプロットし、各自の基準で収束判断することで行ってきた。ゴンペルツ曲線による近似等を行っていないため、品質判断がコストや納期の圧力に影響されるケースもあり、納入後の欠陥密度が組織目標を大きく外れることもあった。今回、以下の施策を実施し、客観的に品質評価できるようにした。

(1) ゴンペルツ曲線、ロジスティック曲線を使った残存欠陥数の予測値を表示

(2) 早稲田大学鷺崎研究室で提案されている予測値の動きによる収束判定

(3) 組織ベースラインとの比較による定量的な品質の評価

また、これらの機能を日常的に使用している文書開発ツールに魅力品質を高めて提供することで、組織展開を容易にした。

<キーワード>

信頼度成長曲線、残存欠陥数の予測値、ゴンペルツ曲線、ロジスティック曲線、ベースライン

<想定する聴衆>

統合テスト時の品質評価を改善したいと考えているマネージャー、開発者、プロセス改善推進者

<活動時期>

2015/2~2016/5

<活動状況>

- 着想の段階 (アイデア・構想の発表)
- 変更を実施したが、結果はまだ明確ではない段階
- 変更の結果が明確になっている段階
- その他 ()

<発表内容>

1.背景

住友電工向けの業務システムを開発する組織であるシステムソリューション事業本部では、毎年、組織目標を設定し、より高い品質を目指して複数の改善活動が並行して進められている。今回は統合テストの改善について報告する。

2.改善したいこと

従来、主観的に行われていた統合テストでの品質評価を客観的に行えるようにする。

(その結果、ソフトウェア納入後の欠陥密度が低下することを期待する。)

3.改善策を導き出した経緯

GQM(Goal Question Metric)手法に関する早稲田大学&IESE&IPA/SEC 共催セミナーに参加した際、早稲田大学鷺崎先生のプレゼン資料の中に信頼度成長曲線に関する提案があった。後日、鷺崎先生を訪問し、その手法を導入させてもらうことになった。また、組織の実力を示すベースラインの確立方法について共同研究することになった。

4.改善策の内容

以下の施策を実施し、客観的に統合テストの品質評価が行えるようにする。

- (a) ゴンペルツ曲線、ロジスティック曲線を使った残存欠陥数の予測値を表示するツールの提供
- (b) 早稲田大学鷺崎研究室で提案されている予測値の動きによる収束判定手法の導入
- (c) 実績値と組織ベースライン（他のプロジェクト）との比較による品質評価

また、改善策の展開が短期間でできるよう、以下の方針で進めた。

- (d) 日常、EVM や管理図で使用している文書作成ツールに本機能を組み込むことで、インストール、基本操作の習得工数を削減する。
- (e) 成長曲線の時間経過をアニメーション表示する機能などで、魅力品質を高める
(鷺崎先生の PowerPoint でのアニメーションが開発者の興味を引いたため)

5.改善策の実現方法

(1) ツール開発

図 1 に示すように、既存の文書作成ツール isdoc と統計ツール R を連携させ、ゴンペルツ曲線やロジスティック曲線の計算が行えるようにした。利用者は、統合テストで得られた、テスト日、検出欠陥数、テスト工数、実施したテスト項目数を CSV 形式でツールに入力する。ツールはこれらのデータを R に引き渡す。R は、予測値の推移を求めるために、初日から 4 日目、初日から 5 日目、…、初日から最終日までといった具合にデータを抽出し、それぞれのゴンペルツ曲線の近似曲線を求め、結果を isdoc に返す。（ゴンペルツ曲線は 3 つのパラメータで示されるため、最低 4 つの測定値が必要となる。） isdoc は図 2 に示すグラフや図 3 に示す残存欠陥数の予測値を HTML 形式の Web ページとして出力する。

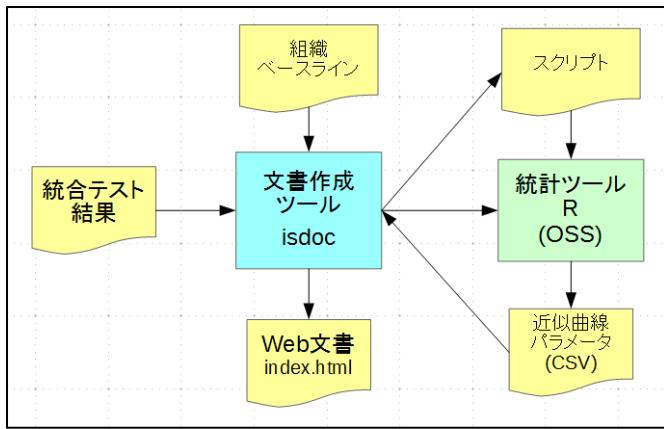


図 1. ツールの構成

図 2 のグラフの横軸はテスト工数の累積値であるが、テスト項目数を横軸にしたグラフも別途表示される。縦軸は欠陥数で、(a) 検出欠陥数の累積値、(b) ゴンペルツ曲線、(c) ロジスティック曲線、(d) ゴンペルツ曲線による予測値の推移、(e)組織ベースラインが表示されている。

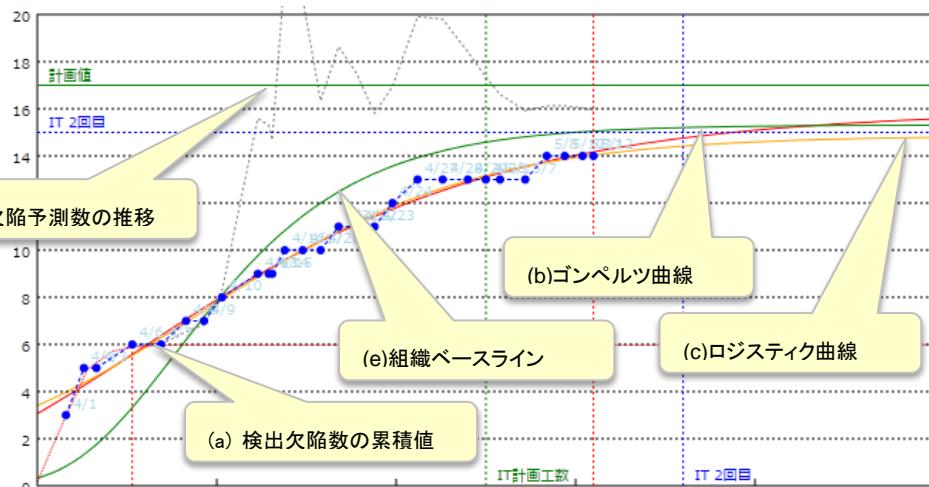


図 2. 実績値と近似曲線の表示

(d)は早稲田大学が提唱している収束判断の手法で利用する。この曲線の変動が大きい場合、まだ収束していないと判断する。また、開発者は組織ベースライン(e)とプロジェクトの実績値である(a)を比較することで、他のプロジェクトに比べ、欠陥の検出量がどの程度多いのか、少ないのかを評価することができる。

また、図 3 に示すように残存欠陥数の予測値を表示している。ゴンペルツ曲線およびロジスティック曲線の近似は時間軸として、テスト工数とテスト項目数の 2 種類の方法で行えるため、合計 4 種類表示している。これらの値が、組織の品質目標を上回る場合、追加テストが必要と判断する。

No.	Item	Value	Item	Value
1	検出数(実績)	14	16 Gompertz残存予測数(工数)	2
2	Gompertz予測数(工数)	16	Gompertz残存予測数(項目数)	2
3	Gompertz予測数(項目数)	16	Logistic残存予測数(工数)	1
4	Logistic予測数(工数)	15	Logistic残存予測数(項目数)	1
5	Logistic予測数(項目数)	15	Gompertz予測誤差	2
6	ST+S+3 欠陥検出数	4		

図3. 残存欠陥数の予測値表示

(2) ベースラインの確立

統合テストの品質評価は、単に信頼度成長曲線が収束するだけではなく、他のプロジェクトと比較して欠陥密度の大きさも評価したい。図4は評価プロジェクトで検出された欠陥を正規化せずに示したものである。規模が異なるため検出される欠陥数やテスト工数、テスト項目数も大きな差がある。

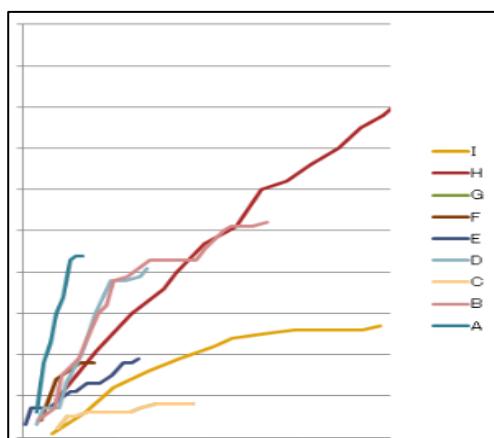


図4. 試行プロジェクトの実績

また、各プロジェクトから得られるゴンペルツ曲線は、式(1)に示すように3つのパラメータ a 、 b 、 c を持つ指數関数であり、どのようにして組織の代表となるゴンペルツ曲線の3つのパラメータを決定するかが課題であった。

$$Y = a * \exp(-b * c^X) \quad (1)$$

試行錯誤の結果、以下の方法で信頼度成長曲線を正規化することで、違和感のないベースラインが求められることがわかった。

縦軸：欠陥密度 = 欠陥数 / ソースコード合計ライン数

横軸1：テスト工数消化率 = その時点でのテスト投入工数 / 全テスト工数

横軸2：テスト項目実施率 = その時点での実施テスト項目数 / IT仕様書のテスト項目数

(テスト項目実施率は、同じテストを2回以上実施するがあるため、100%を超えることが多い。)

ゴンペルツ曲線の近似は実績値に対して最小自乗法を使って求めていることから、全プロジェクトのデータを上記の方法で正規化し、混ぜ合わせて求めた近似曲線をベースラインとして採用することにした。

6. 改善による変化や効果

(1) シミュレーション

開発が終わった 15 プロジェクトのデータをツールに入力し、ツールの評価を行った。その結果、以下のことが判明した。

- (a) 表計算ソフトを使った簡易グラフで収束したと判断したものでも、本ツールを使用すると明確に収束していないと判断できるものが検出された
- (b) ゴンペルツ曲線による残存欠陥数が 10 を超えるものは、納入後、組織の品質目標が達成できていない。
- (c) 稼働後、品質のよいシステムは、図 2 の (b) と (d) の曲線が近づき、平行になっていく。
- (d) 事前の開発者の意見ではテスト項目数の方が当てはまりがよいということであったが、シミュレーションの結果、横軸にテスト工数を採用する方が、当てはまりがよい。

以上の結果から、本ツールの効果があると判断し、全社展開することにした。

(2) 実プロジェクトでの評価

本ツールをリリース後、10 プロジェクトが統合テストを完了した。そのうち 1 プロジェクトは信頼度成長曲線が収束していないため計画を見直し、追加テストを実施した。プロジェクトの担当者から以下の声が届いており、今後、組織の品質目標達成に貢献するものと期待されている。

- ・「欠陥の残存予測数が数字で表示されているので、プロジェクト会議で共有し、あとどれだけテストすべきか共有できる。」
- ・「予測値の推移からより客観的に収束の判断ができる。」
- ・「今までとは全然違う。」
- ・「統合テストの品質評価を自信を持って説明できるようになった。」
- ・「組織ベースラインと比較することで、客観的にプロジェクトの状態が把握できるようになった。」

7.改善活動の妥当性確認

(1) 施策の妥当性

統合テストでの品質評価を客観的に実施できるようにしたいという改善目標は、上記の結果から達成できると考えられる。組織ベースラインとして設定したゴンペルツ曲線も違和感のない納得できる値を示している。ただし、現時点では、実施件数が少ないのでもう少し実績を増やした上で評価する必要がある。

また、ツールは宣伝活動を行わなくても自発的に利用されており、展開活動をスムーズに進める狙いも達成できている。ツールの魅力品質が展開を加速させることができた。

(2) 悪影響の確認

本ツールは、開発作業の中で日常的に利用している文書作成ツールであり、サンプルを示すだけで特別なトレーニングは不要であった。また、「従来よりも楽に作成できる」という声もあり、工数面での悪影響は出ていない。

(3) 今後の課題

今回は、過去 15 プロジェクトのデータで組織ベースラインを仮設定した。今後、定期的に実績データを収集し、毎年ベースラインを改訂する仕組みを運用していく必要がある。

また、統合テストが 5 日程度の小規模プロジェクトではうまく近似曲線が求められないこともあり、対策が必要となっている。

A.参考文献

- [1] 鶴崎 弘宜, 早稲田大学グローバルソフトウェアエンジニアリング研究所 事例報告
ゴール指向の測定によるソフトウェア品質評価と改善の実践的取組み, 2015/2
<http://sec.ipa.go.jp/seminar/20150219.html>