

SPI Japan 2015 発表概要集

- ・セッション順番 > 発表順番に、記載されています。
- ・目次から、各発表の概要にジャンプすることができます。

内容

1A1 「反復開発やアジャイル開発の経験が少ないプロジェクトの推進役を支援するための開発ガイドの作成」 木村めぐみ(オージス総研).. 2	
1A2 「組み込み FW 開発でアジャイルに初トライ！」 深田章敬(ソニー)	9
1A3 「ゼロベースにリセットされたチームと歩むアジャイル支援」 長岡桃子(富士通)	13
1A4 「4人で挑むスクラム DevOps」 清水凌(富士通)	16
1B1 「システム思考アプローチを用いた SPI 手法の提案」 小山貴和子(東芝)	20
1B2 「多様な成熟度の組織をカバーする改善支援サービスの標準化の取り組み」 中村淳(NTT データ)	24
1B3 「現場発ナレッジの全社共有の取り組み」 相澤武(インテック)	27
1B4 「プロセス改善は隗より始めよ～品質保証部の業務改善～」 福本剛(NTT データ)	31
1C1 「静的解析結果の見せる化」 和良品文之丞(キャノン ITソリューションズ)	35
1C2 「設計プロセスの課題を ODC 分析で改善してみた」 小島義也(エプソンアヴァシス)	39
1C3 「メトリクスでソースコードの美しさは判断出来るのか？」 山口祐史(富士フイルムソフトウェア)	43
1C4 「コンカレント開発に向けたアーキテクトチーム導入による開発プロセスの改善」 林健吾(デンソー)	46
2A1 「通信ソフトウェア開発「短期リリース」への道」 兎耳山俊吾(富士通)	49
2A2 「海外開発拠点におけるソフトウェア開発プロセスの確立」 深井十三(Panasonic Industrial Devices Singapore)	53
2A3 「東芝製複合機組み込みソフトウェア開発の海外拠点移管に関する取り組み」 鈴木厚志(東芝テック)	56
2A4 「海外拠点におけるローカル講師育成による品質・プロセス教育の定着」 田村朱麗(東芝)	61
2B1 「埋まっている会社の資産(人材)を最大限に活用する」 長橋敦(シナジーテック)	68
2B2 「7社統合による品質マネジメントシステム(QMS)の再構築」 杉野晴江(NECソリューションイノベータ)	73
2B3 「システムティック類推見積 - 過去事例との新しい出会い」 高橋一郎(日立製作所)	76
2B4 「昔から使ってるので今更変えたくないんだけど」 斎藤学(ウイングアーク 1st)	79
2C1 「システム構築プロセスの標準化と CCSF 活用による IT 人材育成への取り組み」 清水淳史(セイコーエプソン)	83
2C2 「「SQuBOK」と「ワークショップ」を活用した SQA 育成」 菅沼由美子(パナソニック)	86
2C3 「プロセス改善推進者の成長について」 中村伸裕(住友電気情報システム)	89
2C4 「機能安全開発を支える安全文化の確立と実践」 安倍秀二(パナソニック)	95
3A1 「会社が用意したプロジェクト管理ツールを私が使いこなしている2つの要素」 高橋裕之(ウイングアーク 1st)	99
3A2 「Rubyを「やりたい」から「やった」にするまでの道のり」 細美彰宏(日立ソリューションズ)	106
3A3 「ウォーターフォールでスクラム Enjoy！」 松澤将明(富士通ソーシャルサイエンスラボラトリ)	108
3B1 「品質保証部門におけるワーキンググループ活動の活性化を目的とした取り組み」 山本美由紀(日立製作所)	113
3B2 「SPI 経験を活かした SQA 活動実践」 植村貴美(パナソニックヘルスケア)	117
3B3 「なぜなぜ分析におけるファシリテータの課題と改善策の紹介」 林真也(住友電気情報システム)	121
3C1 「約2年半で CMMI ML3 を達成した部門の SPI 活動事例紹介」 小田倉泰浩(東芝デジタルメディアエンジニアリング)	128
3C2 「プロセス改善活動における GQM の評価」 山邊人美(住友電気情報システム)	131
3C3 「自律型プロジェクトチームへの変革アプローチ事例」 安達賢二(HBA)	140

1A1「反復開発やアジャイル開発の経験が少ないプロジェクトの推進役を支援するための開発ガイドの作成」 木村めぐみ(オージス総研)

<タイトル>

反復開発やアジャイル開発の経験が少ないプロジェクトの推進役を支援するための開発ガイドの作成

<サブタイトル>

<発表者>

氏名(ふりがな)：木村 めぐみ(きむら めぐみ)

所属：株式会社オージス総研 技術部 アジャイル開発センター

<共同執筆者>

氏名(ふりがな)：藤井 拓(ふじい たく)

所属：株式会社オージス総研 技術部 アジャイル開発センター

<要旨>

当社では全社的に反復開発とアジャイル開発に取り組んでいるが、反復開発やアジャイル開発が成功するプロジェクトがある一方、成功しないプロジェクトがあった。成功するかはプロジェクトのPM やリーダーの反復開発やアジャイル開発の経験が影響していた。課題は経験が少ないプロジェクトが開発を成功させることである。このため、これまで行ってきたアジャイル開発のトレーニングとプロジェクト支援を見直し、過去の反復開発やアジャイル開発プロジェクトの経験値を開発ガイドという形で蓄積して提供することで経験を補う取り組みを開始した。

<キーワード>

アジャイル開発、反復開発、スクラム開発、トレーニング、プロジェクト支援、プロジェクト測定、規模、開発ガイド、定量的、定性的、横断的

<想定する聴衆>

SEPG、品質保証の方、ソフトウェアエンジニア、PMO、PM

<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(一部変更を実施した。結果はまだ明確ではない段階。)

<発表内容>

1.背景

当社は全社的に反復開発とアジャイル開発に取り組んでいる。反復開発とアジャイル開発は以下のように分類している。

表 1 反復開発とアジャイル開発の分類

	反復開発	アジャイル開発
特徴	・ 一定の期間毎に動くソフトウェアを作る形でソフトウェア開発を進める	・ 反復的な開発を進める ・ 顧客と協調して、可能な限り、ニーズの変化やフィードバックに対応する ・ 開発者の自律性やコミュニケーションを重視する ・ 変化の影響を抑えるためにテスト駆動開発（TDD）や継続的なインテグレーション（CI）などの技術的なプラクティスを適用する
要求の確定度	概ね確定している	要求の確定度：不確定な割合が高い
反復することの狙い	リスク管理	要求の変化への対応

当社では従来手法から反復開発とアジャイル開発に移行することで、不確定な要求や新しい技術への対応力をつけ、付加価値を高めることを目指している。いまのところ、反復開発とアジャイル開発の中間に位置するプロジェクトが多い。

筆者らが在籍するアジャイル開発センターでは、ここ数年アジャイル開発手法のソフトウェア開発の現場への適用を推進するための活動をしており、これまでに以下のような取り組みを行っている。

- ・ アジャイル開発に関するトレーニングの実施
 - アジャイル超入門
 - スクラム入門
 - アジャイル超入門 技術編（TDDとCI）
 - 受け入れテスト駆動開発入門
- ・ アジャイル開発プロジェクトの立ち上げ支援
- ・ プロジェクト測定

アジャイル超入門トレーニングはほぼ全社員が受講し、それ以外のトレーニングは希望するプロジェクトが受講してきた。アジャイル開発プロジェクトの立ち上げ支援については年に2件ほどのペースで行ってきた。プロジェクト測定は、年に4～5件のペースで反復開発やアジャイル開発プロジェクトを測定している。

こうした取り組みを行っている中で、反復開発やアジャイル開発を適用したプロジェクトを見ていると、反復開発

やアジャイル開発を成功させているプロジェクトがある一方、成功していないプロジェクトがあることが分かった。ここで成功とは、表 1 で述べた特徴と狙いを実現しながら、従来手法のプロジェクトと比較して生産性や品質が同等のプロジェクトを言う。

成功していないプロジェクトと成功したプロジェクトのメンバー構成の特徴を見ると、成功していないプロジェクトは、PM やリーダー、開発メンバーに反復開発やアジャイル開発の経験がなく、スクラムマスターがいないか、いても実践経験が少なかった。スクラムマスターはアジャイル開発の一つの手法であるスクラム開発を実施する際に、開発メンバーにスクラムを教えたり、開発が円滑に行えるような土壌を作ったりする役割を持つ。

一方の成功したプロジェクトは、過去に複数のプロジェクトで反復開発やアジャイル開発を行った経験豊富な PM やリーダーがいるプロジェクトだった。このような経験豊富なメンバーが PM やリーダーを務めるプロジェクトでは、経験がないメンバーを育成しながら開発を進めていた。

このことから、反復開発やアジャイル開発の経験がない PM、リーダーや、実践経験が少ないスクラムマスターが率いるプロジェクトでは、開発期間中にメンバーを育成しながら反復開発やアジャイル開発を実践するのは難しかったことが分かった。成功したプロジェクトの例では、反復開発やアジャイル開発の経験が豊富な PM やリーダーが、経験がないメンバーを育成できていることから、これらの成功したプロジェクトを参考にして育成方法を検討した。

まずは反復開発やアジャイル開発の経験が少ないプロジェクトが、反復開発やアジャイル開発を成功させることを目標として、これまでのトレーニングとプロジェクト支援の方法を改善することを計画した。

一番大きな改善点は

- ・ 反復開発やアジャイル開発の経験を補うための開発ガイドをプロジェクトに提供し、実践してもらうことである。

開発ガイドには過去の反復開発やアジャイル開発プロジェクトで起こった問題点と解決策、また取り組むべきよいプラクティスをまとめる。加えて、プロジェクトを定量的に評価できるよう、これまで行っていたプロジェクト測定も行うことにした。

このような取り組みによって反復開発やアジャイル開発の経験が少ないプロジェクトが、反復開発やアジャイル開発を成功させることを目指す。さらには、開発ガイドの提供によって、反復開発やアジャイル開発を進める上でのよくある問題や取り入れるべき点を社内で共有しやすい仕組みを構築することを目指している。

2.改善前の状態

反復開発やアジャイル開発を適用した社内のプロジェクトを見てみると、反復開発やアジャイル開発を成功させているプロジェクトがある一方、成功していないプロジェクトがあることが分かった。

以下に、成功していないプロジェクトと成功しているプロジェクトの特徴を述べる。

- 成功していないプロジェクト
プロジェクトのメンバー構成の特徴
 - ・ PM やリーダーは従来手法の経験はあるが反復開発やアジャイル開発の経験がない
 - ・ スクラムマスターがいないか、いても実践経験が少ない

- ・ 大部分のメンバーは反復開発やアジャイル開発経験がない
- ・ 一部に従来の開発手法に固執するメンバーがいる

問題点

- ・ 早い段階でリスクを顕在化できず後半に問題が発生した
- ・ 反復期間内に開発する意識が薄く、次の反復期間への持ち越し作業が多い
- ・ 反復毎の振り返りで見積り精度や進捗が悪いことが分かっても、改善策が出ず改善されない

● 成功しているプロジェクト

プロジェクトのメンバー構成の特徴

- ・ 過去に複数のプロジェクトで反復開発やアジャイル開発を行った経験豊富な PM やリーダーがいる
- ・ 大部分のメンバーは反復開発やアジャイル開発経験がない
- ・ プロジェクトで反復開発やアジャイル開発経験のないメンバーをサポートするための仕組み（Wiki での情報共有、開発ガイド）を用意している
- ・ 改善の意識が高い

よかった点

- ・ 反復開発を進めながら変更要求に対応した
- ・ 前の反復の問題点を分析し、改善策を検討し実際に改善を実施した
- ・ 改善によって品質や生産性が向上した

3.改善前の状態をもたらした原因（因果関係）

前の章で挙げた、成功していないプロジェクトのメンバー構成と問題点は以下のとおりである。

プロジェクトのメンバー構成の特徴

- ・ PM やリーダーは従来手法の経験はあるが反復開発やアジャイル開発の経験がない
- ・ スクラムマスターがいないか、いても実践経験が少ない
- ・ 大部分のメンバーは反復開発やアジャイル開発経験がない
- ・ 一部に従来の開発手法に固執するメンバーがいる

問題点

- ・ 早い段階でリスクを顕在化できず後半に問題が発生した
- ・ 反復期間内に開発する意識が薄く、次の反復期間への持ち越し作業が多い
- ・ 反復毎の振り返りで見積り精度や進捗が悪いことが分かっても、改善策が出ず改善されない

上記の問題は以下のことに起因したと考えている。

- ・ 潜在的なリスクがあることを想定せず反復計画を立てた
- ・ プロダクトバックログの完了条件があいまいなまま開発を進めた
- ・ 反復計画で開発メンバーが実施するタスクの粒度が大きく、タスクがなかなか完了しなかった
- ・ リリース計画や反復計画の進捗をバーンダウンチャートで可視化しなかった

上記の原因によって、タスクの進捗や、見積り作業時間と実績作業時間の乖離、残作業時間や残ポイント数と言ったプロジェクトの状況が把握できず、問題点が明確にならなかった。問題点が明確にならなかったため改善策も出なかったのだと推測している。

上で挙げた原因はアジャイル開発で基本的な実施すべきことであり、基本的な概念はトレーニングで伝えている。しかし、トレーニングを受けただけでは実プロジェクトでこれらを実践するのが難しかったことが分かった。このことから、トレーニングに加えて、トレーニングで習得した概念を実プロジェクトで実践するための何らかの支援のステップが必要だと認識した。

4. 計画した変更内容

前の章で述べたように、反復開発やアジャイル開発で基本的な実践すべきことが実施できなかったのは、トレーニングで概念を理解しただけでは実践に結び付けるのが難しかったためだと考え、これまでのトレーニングに加え、トレーニングで習得した概念を理解した後、実践のために必要なノウハウを習得するステップを追加することを計画した。

[概念] 反復開発やアジャイル開発の基本的な概念を理解する

↓

[実践] 反復開発やアジャイル開発を実践する上で起こり得る問題点を把握して、実プロジェクトで問題の解決策を実施する

[概念] はこれまでも実施していたアジャイル開発に関するトレーニングを実施することで実現する。対象はすべての開発メンバーである。

[実践] は新たに計画した内容である。これまでに反復開発やアジャイル開発を実施したプロジェクトの経験を元に、注意すべき点や取り組むべき点を開発ガイドの形でまとめる。開発ガイドはスクラムマスターや PM、リーダーに説明し、説明を受けたそれぞれのプロジェクトが問題の解決策を実施することを計画している。

改善の取り組み全体としては以下を行う。

- 1) 経験がないメンバーに対してトレーニングを行う（既存の取り組みの継続）
- 2) 過去のプロジェクトの経験を元に開発ガイドを作成してプロジェクトに提供する（新しい取り組み）
- 3) 開発ガイドを用いて各プロジェクトのスクラムマスターを支援する（既存の取り組みの改善）
- 4) プロジェクトを測定して定量的に評価する（既存の取り組みの改善）

これらを組み合わせて実施することによって、問題をできるだけ解決し、反復開発やアジャイル開発の経験が少ないプロジェクトも成功する確率を高めることができると期待している。

5. 変更の実現方法

- 1) 経験がないメンバーに対してトレーニングを行う（既存の取り組みの継続）

これまで実施していたアジャイル開発に関するトレーニングを継続して実施する。必要なプロジェクトにトレーニングを提供できるよう、社内での広報活動にも力を入れる。

- 2) 過去のプロジェクトの経験を元に開発ガイドを作成してプロジェクトに提供する（新しい取り組み）

これまで関わったプロジェクトを分析して、アジャイル開発を成功させるための開発ガイドを作成する。開発ガイドは社内説明会で説明し、さらに、トレーニングやプロジェクト支援の場で活用することで、実プロジェクト

での適用を促進する。

開発ガイドの対象読者はPMやリーダー、または、スクラムマスターを想定している。これらの読者が開発ガイドを理解し、プロジェクトの中で実践していくことを期待している。

開発ガイドの内容は、まずは反復開発やアジャイル開発の経験が少ないプロジェクトを対象とした内容に重点を置き、次に経験に関わらず活用できる内容を載せ、少しずつ充実させることを計画している。

開発ガイドには、以下の内容を取りこむ。

- ・ 過去の反復開発、アジャイル開発プロジェクトで明らかになった注意すべき点、取り組むべき点

また、これまでのプロジェクト測定で、影響度の大きい問題が複数プロジェクトに共通して起こっていることが分かったことから以下の内容も含める。

- ・ 定量的なプロジェクト分析から明らかになった複数プロジェクトに起こり得る問題点と改善策

説明会では開発ガイドの説明に加え、ケースメソッド形式の演習を行い、過去のプロジェクトを事例に問題の解決策を検討して理解を深めてもらうことを考えている。

3) 開発ガイドを用いて各プロジェクトのスクラムマスターを支援する（既存の取り組みの改善）

初めて反復開発やアジャイル開発を行うプロジェクトのスクラムマスターは、開発メンバーを育成しながらプロジェクトを支援するが、支援内容はこれまでスクラムマスターの経験に任せるところが大きかった。

新しい取り組みでは、過去のプロジェクトの経験を元に作成した 2)の開発ガイドをスクラムマスターに提供し、経験のないプロジェクトに対して支援すべき内容をスクラムマスターに的確に伝える。このことで、スクラムマスターの経験に関わらず、プロジェクトを支援できることを目指している。

アジャイル開発センターは、社内説明会や個別説明で開発ガイドの内容をスクラムマスターや PM、リーダーに説明し、そのプロジェクトに必要なプラクティスを実践してもらうことを目指す。

4) プロジェクトを測定して定量的に評価する（既存の取り組みの改善）

これまでのプロジェクト測定分析では測定対象のプロジェクトについて分析することに力をいれていた。今回の改善では、これまでの測定対象のプロジェクトの分析に加え、過去に測定した複数プロジェクトも含めて横断的に分析し、共通する問題点を見つけて原因を分析するステップを追加する。ここで明らかになった内容は 2)の開発ガイドに取りこんでいく。

6. 変更後の状態や改善効果

以下の内容は実施済みで、今後も継続して実施する。

- 1) 経験がないメンバーに対してトレーニングを行う（既存の取り組みの継続）
- 4) プロジェクトを測定して定量的に評価する（既存の取り組みの改善）

以下の内容はこれから実施する。

- 2) 過去のプロジェクトの経験を元に開発ガイドを作成してプロジェクトに提供する（新しい取り組み）
- 3) 開発ガイドを用いて各プロジェクトのスクラムマスターを支援する（既存の取り組みの改善）

7.改善活動の妥当性確認

トレーニング、開発ガイド、プロジェクト支援を組み合わせることによって反復開発やアジャイル開発の成功確率が高まることを期待している。この改善活動は取り組みはじめたところだが、今後、改善した内容の効果を評価していきたい。

1A2 「組み込み FW 開発でアジャイルに初トライ！」 深田章敬(ソニー)

<タイトル>

組み込み FW 開発でアジャイルに初トライ！

<サブタイトル>

～開発現場から生の声をお届けします～

<発表者>

氏名（ふりがな）：深田 章敬（ふかた あきたか）

所属：ソニー株式会社 デバイスソリューション事業本部

<共同執筆者>

氏名（ふりがな）：牧野 順（まきの じゅん）

所属：ソニー株式会社 デバイスソリューション事業本部

<要旨>

アプリケーション開発やクラウドサービスではアジャイルのイテレーション開発の親和性が高く、決まったリズムでテンポよく開発していくことが容易に可能であるが、HW 製品ありきの組み込み SW 開発において、それを実行するのはなかなか難しい。しかし、HW 製品開発であっても、特に新製品開発では開発途中での要件変更や顧客デモとそのフィードバックを反映していくことが現実として頻繁に行われる。よって組み込み SW 開発においてもアプリ開発と同様にアジャイルの考えを取り入れていかなければ、要望に応じた成果物を繰り返しリリースすることができない。本発表はそのような背景を元に従来のウォーターフォール式からアジャイル式の開発スタイルに初めてトライした時の内容である。メーカーでの HW 主導の製品開発でよく起こるエピソードをベースに、組み込み SW 開発におけるアジャイル開発について議論をしたい。

<キーワード>

アジャイル、スクラム、かんばん、組み込み、ファームウェア、バックログ、スプリント、反復開発、イテレーション、バーンダウン、ヴェロシティ、朝会、振り返り

<想定する聴衆>

組み込み開発 SW エンジニア、SEPG、PMO

<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：HW 製品の FW 開発チーム
- アジャイル導入に至った経緯・・・これまでのウォーターフォール式の所感
 - 顧客要求・要件の確定が遅い
 - ◇ 「要求分析・要件は、まだ流動的だよ」→3ヶ月
 - 状況変化への適応速度が遅い
 - ◇ 「市場動向の変化でXX機能を増やしたい！！」→仕様分析・設計に入っていないので大きな設計・実装変更になるので多くの時間がかかり、機能自体を提供するのにさらにかかる。
 - 顧客・機能のフィードバックが遅い
 - ◇ 「XX機能をすぐ見てフィードバックしたい！！」→実装完まで待ってください。
 - 人の流動への対応が難しい
 - ◇ 「設計まで実施したAさんが離れ、Bさんが担当となる」→設計内容をすべてBさん引き継いで、実装を進める。引き継ぎ優先度は、なし。
 - 緊急対応能力が低い
 - ◇ 「来週、XX機能の状況を見たい！！デモみたい」→まだ設計中なのでまだ見せられません。1ヶ月まってください。
 - ◇ 「A機能入れてよ！！」→仕様分析・設計に入っていないので大きな設計・実装変更になるので多くの時間がかかり、機能自体を提供するのにさらにかかる
 - 最終的に・・・
 - ◇ トータル開発速度が遅くなる。時代遅れになる。顧客の要求を満足できない

2.改善前の状態

- ◇ 上記「1」で記述した様々な問題や症状
- ◇ チーム力の向上・発展
- ◇ 顧客との協調
- ◇ 動作するFWの早期提供
- ◇ 変化への素早い対応

3.改善前の状態をもたらした原因（因果関係）

- ◇ 最終成果物が顧客の期待と違う事がある。要件内容の認識がズれていた…。そもそも顧客の要件も曖昧だった…。
- ◇ PJ途中の変更要求はNG。初期段階で要件や仕様が固まっていることが前提。
- ◇ PJ終盤の品質リスクが高い。前工程からの潜在不具合が、終盤のテスト工程で一気に出る可能性…。
- ◇ 改善サイクルが遅い。PJが終わってから振り返りをしても遅い。
- ◇ これまで携わったプロジェクトの振り返りを行った。

4.計画した変更内容

- ◇ 新開発手法としてアジャイルの導入、スクラム体制・環境の構築
- ◇ 期待効果に対して、コンセプトが合致する開発手法だと考えられたため。
- ◇ 既存の開発手法（ウォーターフォール式）に限界を感じており、ちょうど新製品の開発にアサインされたため、新しいメソッドに挑戦しようと考えたため。

5.変更の実現方法

- アジャイルおよびスクラムの概要・用語を理解、メンバーに説明（Wikiにも記載）
 - アジャイルの考えをベースとした開発ルールの策定と周知
 - スクラムプロセス、およびスプリントイベントの Criteria（達成事項）を策定
 - Redmine と Backlogs Plugin によるバックログ、スプリント、かんばん、バーndダウンの活用
 - ヴェロシティの計測
 - 顧客や Executive デモから積極的にフィードバックを獲得
-
- ◇ HW 開発スケジュールやデモ日程に合わせてスプリントの期間を毎回調整することで、必要な機能を必要な時期にリリースできるよう工夫した。
 - ◇ ビジネス状況や HW の開発状況に依ってスプリント期間内に急遽優先度の高い FW 要件追加・変更が入り、スプリント当初に決めた要件の未完了やリスクが頻繁に発生した。スプリント期間内の変更は抑えられないため、HW 側や海外パートナーなどステークホルダーと積極的に会話をするこゝで、要件の優先度を常に明確にしておくようにした。
 - ◇ 朝会は、朝遅く来るメンバーがいたり、休み人がいたり、朝一で別 MTG が入ったりして、開催に難があった。昼会や夕会を開催するなど工夫した（が、最終的には Daily の開催は難しかった）。
 - ◇ かんばん上で全体の要件を一目で把握したいため、ツール上のバックログの利用は止め、すべての要件をかんばんに載せて運用した。

6.変更後の状態や改善効果

- ◇ HW と協調して機能を積み上げ、顧客にリリースし、フィードバックを次につなげる開発を実現。
 - ◇ 最終的に安定した品質の中でリファクタリングまで実現。
 - ◇ →目標(期日・品質)を達成
 - ◇ →チーム力(組織力)向上
-
- ◇ チーム力の向上・発展⇒スプリント毎の振り返りなどによる改善を重ね、メンバー変更が発生してもパフォーマンスが落ちないチームを構築
 - ◇ 顧客との協調⇒顧客や Executive へのデモ & フィードバックを数多く実現
 - ◇ 動作する FW の早期提供⇒HW と協調して実行可能な機能を段階的にリリースしていくことを実現
 - ◇ 変化への素早い対応⇒開発途中で急遽変更要求が入っても、すぐにそれらを優先して開発できる体制を確立。

- ◇ メンバー変更でもヴェロシティを維持
- ◇ 多数の顧客デモに対応
- ◇ FW 起因のリリース遅延ゼロ
- ◇ 度重なる変更要求に対応

7.改善活動の妥当性確認

- ◇ 妥当性
 - メンバ間の一体感が高まる
 - チーム力(組織力)が一気にあがる
 - 方向性そろえられる。方向性調整が容易。
 - スピード重視・変更適応能力高い。
 - リファクタリング・変化当たり前。
 - 人の入れ替わりにも対応しやすい
- ◇ 費用対効果
 - スクラムプロセスを回すためのツールはフリーツールの Redmine と Backlogs plugin を利用。
 - かんばんの利用で進捗管理が効率化。(工数減)
 - コストはかからず(むしろ減った中で)、期待効果が出たという点で、費用対効果は高かった。
- ◇ 残存課題
 - 成果の個人依存性が高い。アジャイルはソースコードのスパゲッティ化が避けられず、ある程度成熟したメンバーが必要。
 - 各スプリントで対応する要件は自分達の言い値なので、サボろうと思えばいくらでもサボれる。
 - HW スケジュールの影響が大きい。組み込み FW 開発で反復期間を一定にすることは難しい。
 - アジャイル開発に対する周囲の理解が不足。(一時期製品の遅れを FW 開発の遅延のせいになされてしまった。実際には優先度の高いデモ要件を優先して開発したため、残件が後ろのスプリントに回っただけ)
- ◇ 副次効果
 - リファクタリングを行うスプリントを設けたことで、品質が向上した。
 - 大規模開発への展開もあり。例えば 40 人規模であっても、導入可能と思われる。各チーム編成を 5 名程度にするのがベスト。各チームリーダーが同一思想を持っていないとだめ。チームリーダーが一つのチームだという認識を持ってもらう必要がある。

1A3 「ゼロベースにリセットされたチームと歩むアジャイル支援」 長岡桃子(富士通)

<タイトル>

ゼロベースにリセットされたチームと歩むアジャイル支援

<サブタイトル>

チームづくりから社内キーマンのコミュニティづくりへ

<発表者>

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

<共同執筆者>

氏名（ふりがな）： 松浦豪一（まつうらひでかず）

所属：株式会社富士通マーケティング

氏名（ふりがな）： 永井美雪（ながいみゆき）

所属：富士通株式会社

<要旨>

様々な企業において「変化、フィードバックに対して柔軟に速く対応できる、アジャイルなチーム」を社内に作り、増やすことが求められている。富士通では、現場支援者が、支援対象チームにとって成長しやすい場づくりを行うことでチームづくりを支援してきた。慣れたやり方を捨てさせるためのノウハウや、成長を支援する心構えを発表する。

<キーワード>

アジャイル、スクラム、現場支援、教育、プロジェクトファシリテーション

<想定する聴衆>

高成熟度組織の方、SEPG

<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

ビジネススピードにマッチしたものづくりへの関心が高まっており、アプローチの一つとしてアジャイル開発を適用する企業が増えている。富士通も例外ではない。富士通は SI 部門でアジャイル実践センターを設立し、ワークショップ型の実践教育と現場支援を通して 2011 年からアジャイル実践ができるチームを作るノウハウを蓄積している。教育での疑似体験と有識者による現場支援から、アジャイル実践できるチームを作り、増やしたいと考えている。教育や支援を活用しながらアジャイルを習得し成果を上げた社内事例が出てきている。プロダクト部門でもアジャイル WG が立ち上がるなど、全社でアジャイルへの関心が高まっている。

2.改善前の状態

まず、あるべき姿である「変化、フィードバックに対して柔軟に速く対応できる、アジャイルなチーム」とは、プロセス、マインドセットが硬直しておらず、自己組織化（立場関係なく、色々な人が色々なタイミングで判断ができる）チームであると考え。そのようなチームを目指すべく、アジャイル適用を試みるものの、アジャイルの良さを生かすことなく適用を断念し、元の慣れたやり方に戻ってしまうケースが散見されていた。慣れたやり方とは、細かく計画し、ズレを管理するウォーターフォール(WF)型のスタイルである。WF 型においては、変更は不安要素であり、ドキュメント（手順書・設計書・チェックリスト）が重要となる。アジャイルのノウハウを書籍など形式知だけから習得しようとし失敗するケースもあった。開発作業そのものもドキュメントに依存しており、設計書が無いと開発ができない、と語る開発者もいた。また、「アジャイルは速く開発できる」「アジャイルは人を選ぶ」などの誤解や、そもそも富士通のような日本の大企業にアジャイルは合わない、との声も多く、展開の阻害要因となっていた。

3.改善前の状態をもたらした原因（因果関係）

1) 慣れたやり方を捨てることなくアジャイル適用を試みていたこと。変更への不安と上意下達チームに染み付いていた。2) 身近なお手本がいなかった。

4.計画した変更内容

現場支援において、それぞれのチームのマインドセットや慣れたやり方を「リセット」することを重視して活動した。実践者の社内の交流や、社外発表を促進した。

5.変更の実現方法

教育や、定期的な現場支援において、書籍から理解できない暗黙知の共同化を促進するしくみを示した。具体例は、ミーティングの時に、ボードの前に、隣の人と肩が触れ合うほどの距離で立って集まることである。これにより、リーダー以外の発言のハードルが下がり、物理的に問題 VS 私たちの構図を作り出し人と問題を切り離す（プロジェクトファシリテーションより着想）ことができる。このように、それまでと違ったやり方を持ち込み、それまでのやり方を捨てさせ、「リセット」させ続けた。またコミュニティ作りとしては、有識者による現場支援時に別のアジャイルチームにも参加を依頼して交流を促したり、JASPIC 例会での外部講演のセッティングや、読書会を行った。方法論の導入ではなく、環境作りを念頭におき活動した。

6. 変更後の状態や改善効果

現場支援において前述のような訓練を受け、リセットされた状態から成長したアジャイルチームが、成果を上げた後に社内外で事例を発表した（2014年度は4事例）。とあるチームは、アジャイル実践ができるようになったが、教育を早い段階で受講せず、チームビルディングに時間を要してしまったという反省点がある。別のチームは、コマンド&コントロール文化から愚直な工夫を重ね、自律的なチームになっていったという教訓がある。彼らの体験談が生きたノウハウとなっており、後から続くアジャイル初心者達に広まっている。また、互いに困った時に頼り合える、実践者同士のコミュニティが出来つつある。

7. 改善活動の妥当性確認

アジャイル適用時に、慣れたやり方を捨てさせるために、現場支援で意識すべき点がわかった。副次的な効果として、社内外に発表を行える富士通のアジャイルキーマンを育成できた。今後このようなキーマンを増やしていくには、キーマンのためのコミュニティをデザインすることが求められる。その際コミュニティはあくまで経験からの暗黙知を共同化することに注力する場であり、それが富士通でのアジャイル展開の鍵となると考える。手順書やノウハウ集を最終アウトプットとせず、生きたノウハウを進化させ続けることを重視することで、変化に柔軟に速く対応するチームを増やしていきたい。

1A4 「4人で挑むスクラム DevOps」 清水凌(富士通)

<タイトル>

4人で挑むスクラム DevOps

<サブタイトル>

社内サービスの早期デプロイ目指して早1年

<発表者>

氏名(ふりがな)： 清水 凌(しみず りょう)

所属：富士通株式会社

<共同執筆者>

氏名(ふりがな)：

所属：

<要旨>

社内サービスの品質向上のためスクラム開発を始めたところ、モノが出来上がる速度が格段に向上した。だがデプロイの速度が間に合わず、利用者へ公開されるのが2週間後ということが頻発していた。そこで DevOps にトライしてみたところ、利用者公開は格段に早くなった。しかし次は開発の速度が遅くなってしまった。本論では、スクラムマスターではなく開発チーム2年目のひよっこメンバーの視点で、スクラム開発の中で DevOps を実践してみたの感想・問題点を発表する。

<キーワード>

アジャイル開発、スクラム開発、DevOps、継続的デリバリー、継続的インテグレーション

<想定する聴衆>

アジャイル開発に関わる方、DevOpsに興味のある方

<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

社内向けシステムの開発をスクラム開発に取り組み始め、成果物が出来上がるペースは格段に早くなった。しかしいくら早くなると運用環境へのリリースが遅ければ、ユーザにとっての価値はあまりない。本来、スクラム開発を代表とするアジャイル開発手法は、リリースの時間短縮と頻度を増加させるためのアプローチである DevOps との相性が非常に良いとされている。スクラム開発になれてきたチームのレベルを上げるため、また利用者への早期提供のため、DevOps に取り組むこととなった。

2. 改善前の状態

私が開発チームに加わった当時は、システムの開発と運用は部隊が分かれており、それぞれがどのような作業を行っているか、リーダーレベルでも詳しくは把握できていない状態であった。よって、いくら開発スピードが上がろうと運用部隊はそのことに気づくことが無く、運用環境へのリリースは以前から変わらず2ヶ月に1度ある定期メンテナンスのタイミングのみ。スクラム開発手法を採用するきっかけとなった開発チームリーダの『より良いサービスを“素早く”提供したい』という思いからは、まだ程遠い状況であった。

その思いを実現するべく、DevOps の文化を取り入れるに至った。

3. 改善前の状態をもたらした原因（因果関係）

各部隊の作業を表してみると、弊部で開発しているシステムには以下の部隊が関わっている構造となっていることが分かった。

- 開発部隊：全8サービスのアプリケーション開発および保守。4人体制。
- システム基盤運用部隊(他PJ)：全サービスの動作する、ハード～ミドルウェアの構築・保守。
- 各サービス運用部隊(他PJ)：各サービスの運用およびインフラ構築・保守。
- サービス統括部隊(他部署)：全サービスの方針決定、利用者ニーズの把握、利用課金計算など。

上記の通り、運用部隊が2つ存在し、上位にサービス全体を取りまとめるサービス部が存在している。これらの部隊と開発部隊の関係性は、以下の通りである。

- I. サービス部およびサービス運用部隊からの要望を元に各サービスのアプリケーションを作成。
- II. 出来上がったものをサービス部およびサービス運用部隊に見せ、レビューを行う。
- III. サービス部からのGOサインが出れば、システム基盤運用部隊に渡し、本番環境へデプロイしてもらう。

ここで問題視したのが、III.のプロセスである。開発部隊としては、テストも完了し問題が無いと判断したので早く運用環境へデプロイし、利用者に公開してほしい。しかしサービス部・サービス運用部隊の立場としては、利用者への影響はなるべく抑えたいため、休日に行う定期メンテナンスのタイミングで本番配備を行い、十分検証を行ったうえで利用者に公開したいという思いが強い。つまり、各部隊ともに“良いサービスを提供したい”という思いは同じであったが、開発部隊とは“素早く”という部分に意識の差異があった。開発チームとしては、この意識の差が運用環境への早期デプロイを阻害している原因ではないかという考えに至った。

4. 計画した変更内容

開発部隊がスクラム開発に取り組み始めて1年が経ち、早期に成果物ができるようになってきていた。そこで『より良いサービスを“素早く”提供したい』という目標により近づくべく、アジャイル型開発と相性が良いとされている DevOps というアプローチを採用した。DevOps とは、一言でまとめると開発・運用の協力体制の在り方を示すものである。明確な定義は存在しないが一般的にこの手法を採用することで、開発・運用部隊それぞれのサイロ

化を解消し、リリースおよび本番環境配備の頻度の増加が見込めるとされている。この DevOps の考え方に従い、以下の 2 点の実現を目指した。

- 運用と開発の壁を取り除き成果物の運用環境への配備頻度を増加させることで、成果物の素早い提供を行う。
- より利用者へ近い立場となることで利用者ニーズの把握を容易にし、より良いサービスの提供を行う。

5. 変更の実現方法

DevOps において、開発・運用部隊それぞれが相互に理解を深めることは非常に重要とされている。しかし運用部隊に対していきなり「開発してください」とお願いするのは難色を示されてしまう。そこで、まずは開発部隊から運用に関わっていき、という方向性で解決を試みた。協議の結果、システム基盤運用における OS より上位層、つまり OS、ミドルウェア、アプリケーションの運用・保守を、開発部隊が担う形となった。また利用者ニーズの把握と、利用者からの問い合わせ回答時間短縮のため、利用者対応も行うようにした。施策の実施当初は、初めて利用者とは直接やり取りする、本番環境に直接乗り込む等、ノウハウもない慣れない作業ばかりで苦労することが多かったが、数をこなせば慣れてきた。しかし慣れてきたゆえに油断も生まれ、本番環境へ旧版のバイナリをデプロイしてしまうという問題を起こしてしまうこともあった。

上記の施策のみであると、開発部隊の負担のみ増加してしまい、開発部隊が少人数であることと相まって、運用作業が積み重なった場合には開発が止まってしまうという最悪の事態も考えられる。そこで開発自体を効率化するために、開発プロセスにも手を加えることにした。具体的には、結合テスト実施時にサービス運用関係者を呼び、利用者視点でテストを行ってもらおうと同時にリリース判定も行ってもらおう。その際、本番配備における影響範囲等を話し合い、既存サービスに対しての影響は軽微だと判断した場合には、即時本番環境へのデプロイを行うことにした。弊部はデータセンターという特殊な立ち位置であるため、テスト環境用のリソース等は比較的恵まれており、本番環境と同様のテストが可能であったため、上記のような手法を取ることができた。

施策を実施中、開発部隊メンバーが 4 人に減少するという事態が発生(スクラムマスター含む)。しかし運用作業が減るということは無いため、開発作業が割を食うことになってしまう。そこで今回の施策では、ベロシティを確保するためにスクラムマスターが開発作業へ加わるという、通常のスクラムから離れた状態で開発を進める場面を設けた。結果として、現在も 4 人体制であるが開発は滞りなく回っている。

6. 変更後の状態や改善効果

開発部隊が運用を、運用部隊が開発にも加わるようになったことで、本番環境へのデプロイリスクの低下を実現、本番環境へのデプロイ回数を施策実施前の 4 倍近くまで増加することができ、結果として利用者満足度の向上にも繋げることが出来た。

それぞれの施策別に効果を報告する。

OS より上位層を開発部隊で運用・保守を行うことで、何かしらの問題発生時に早期発見・解決を達成できた。また新規にサービスを始める場合には、開発部隊で選定した OS やミドルウェアの構成をそのまま本番環境へ適用できる、というメリットが発生。これにより、新規サービス立ち上げのコスト・リスクが大幅に低下したと言える。現在は運用部隊と連携してシステム監視部分周りのプロセス改善に取り組んでおり、今後はトラブル予測などへ繋げていきたい。利用者対応を開発部隊でも行った結果、利用者トラブル解決までの時間を短縮することに成功し、また利用者の生の意見を聞く機会が増えたことにより、より利用者ニーズに沿ったサービスの開発に努めることに成功した。

結合テスト時にサービス運用関係者と共にテストを実施することで、仕様要件とのズレを早期に発見すること

が可能となり、更なるリリース速度向上を果たすことができた。また本番環境と同様のテスト環境を用意することが出来たため、本番環境へのデプロイリスクの低下にも繋げることができた。

スクラムマスターが開発作業に加わるという点では、その副産的な効果だと思いが開発部隊のメンバー全員がスクラムマスターと同じようにサービス部や運用部隊との調整や問題の分離・解決を行うスキルを得ることができ、細かなプロセス改善や開発環境整備などを各員の判断で行えるようになった。

7.改善活動の妥当性確認

運用部隊との距離が縮まったことで、スクラム開発とはどういったものであるのかの理解を得ることができた。運用部隊も呼んで結合テストを実施するという施策も、当初はあまり乗り気ではないようだったが、何度か数をこなす内にメリットを実感してもらえるようになり、現在は積極的に参加して貰えるようになった。だが、開発と運用の壁が全て取り払われたという、決してそうでは無いだろう。むしろ一定の線引きは必須であると感じる。極端な例を挙げれば、開発がインフラ構築にまで手を出し始めると、もはや開発が回らなくなるのは目に見えて明らかだ。さらにそのインフラを運用部隊が管理となった場合、仮にインフラ部分で問題が発生した場合、開発・運用部隊どちらが責任を持つのか、という事態になるだろう。つまり、開発・運用のそれぞれが、どこまで関わり合うかのバランスが重要だ。今回の施策においては、開発・運用部隊の相互理解についての効果を実感し、また利用者への提供の速度向上が実現したという点を考慮すると、開発・運用の関わり程よくバランスを保つことができ、DevOps というアプローチでの解決は成功を取めたと判断できる。しかしそのバランスを保つため、開発部隊の負担が大きくなりすぎたことは確かである。

先に述べている通り、開発部隊が4人という少数規模となってしまう、開発と運用をこなす為には既存のスクラム開発とは異なる方式を取らざるを得ないことがあった。スクラムマスターが開発に入ることで発生する穴(対外調整、問題の分離等)を、各メンバーである程度補うことができたからこそ、今回の施策では少人数でも開発・運用の両立ができ、結果的に目標達成へ繋がったのではないだろうか。

目標は達成できたものの、スクラムマスターの仕事を埋めきれずに新たな問題が発生している。項目5にて述べている、本番環境への配備方式だ。本来であれば、スクラムマスターがこの問題を解決するための糸口(Chefを代表する構成管理ツールや、本番環境デプロイプロセスの改善)を見つけ、開発の影でこっそりと進めるのが正しい在り方だろう。チームメンバー全員とも本問題を意識してはいるが、現状に変わりはなく目視確認のみである。このような規模の大きな改善活動に取り組めていない状況は好ましくないが、今のところ手は打てずにいる状況だ。

私のチームを取り巻く環境は、社内システムを作っているという点(顧客が社内にいる点)や、テスト環境のインフラが充実していたからこそ、成功を取めることが出来たのかもしれない。つまり、世間一般でよく紹介されているアジャイルプラクティスとは大きく異なっている。そのため、今回紹介した施策がどこでも通用するとは言い難いだろう。しかし私たちのいる環境は、新たな取り組みに挑戦しやすい、チャレンジングなことが可能な環境とも言える。この利を生かし、アジャイル開発についての新たな取り組みを今後も行っていきたい。

1B1 「システム思考アプローチを用いた SPI 手法の提案」 小山貴和子(東芝)

<タイトル>

システム思考アプローチを用いた SPI 手法の提案

<サブタイトル>

<発表者>

氏名（ふりがな）：小山 貴和子（こやま きわこ）

所属：株式会社 東芝

<共同執筆者 1>

氏名（ふりがな）：艸薙 匠（くさなぎ たくみ）

所属：株式会社 東芝

<共同執筆者 2>

氏名（ふりがな）：田中 史朗（たなか ふみあき）

所属：北芝電機 株式会社

<要旨>

SPI 活動において、Plan-Do-See(PDS)の流れにシステム思考アプローチの考え方を当てはめて実践することを提案する。システム思考アプローチを導入することで、課題を俯瞰的・多様な視点から捉え、全体像を様々な要素の繋がりのシステムの構造として表すことが可能となり、全体最適化や本質的な問題解決を図っていくことが期待できる。システム思考アプローチでは定量的な変化の因果を描き表すため、改善時の成果効果を定量的に把握することが可能になると考えられる。

<キーワード>

SPI, システム思考アプローチ, ループ図, レバレッジ・ポイント, 定量的な成果効果, ロジカル・シンキング, 課題の分析

<想定する聴衆>

SEPG リーダ, SQAG リーダなど SPI の活動計画立案に関係する方

<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

一般的な SPI 手法では、改善活動を診断・改善計画フェーズ(Plan)、改善策の検討・実行フェーズ(Do)、改善成果の評価フェーズ(See)という流れで進めていく。SPI では改善策の効果が実行からすぐに出るものもあれば、遅れて効果が出る(タイムラグがある)ものもあるため、改善の評価が困難な場合がある。改善の評価が困難な場合があるため、現場には改善の効果を実感してもらって、効果を把握しやすいようにすることが重要である。効果を実感してもらうためには、改善の効果が大きいものを狙って取り組むとよい。そのためには、Plan フェーズの診断で課題の洗い出しと、洗い出した課題に対する分析が重要になる。

2.改善前の状態

診断の方法として、トップダウン型のモデルベースアプローチとボトムアップ型の課題ベースアプローチがある。トップダウン型のモデルベースアプローチは CMMI(Capability Maturity Model Integration: 能力成熟度モデル統合)や ISO9000 などのプロセス改善モデルと照らし、対象組織の強みと弱み(改善点)を把握する方法である。一方、ボトムアップ型の課題ベースアプローチは対象組織が抱えている問題や困っている状況を現場から直接吸い上げて課題を把握する方法である。どちらの方法もあるべき姿と現状のギャップから課題を洗い出すという点では共通している。

さらに、洗い出した課題の原因を分析する際に、論理ピラミッドやロジックツリー、因果関係図、特性要因図、マインドマップ、KJ 法、ブレインストーミングなどのロジカル・シンキングの分析ツールを使う場合がある。これらの分析ツールを使うことで因果などの関係性を明確にすることが可能となる。特に、ロジカル・シンキングを用いた分析ツールでは課題の原因を深掘して追究できる点が優れている。しかしながら、課題を俯瞰的に捉えたり、複数ある課題の相互関係や他の課題からの影響など複雑な関係性を表したりすることが難しい場合がある。そのため、個別最適や局所最適な解決策を打ってしまい、その結果、改善しても効果が十分に得られないことがある。

3.改善前の状態をもたらした原因(因果関係)

ロジカル・シンキングの分析ツールでは、課題の原因を深掘して分析してしまうため個別最適や局所最適な解決策を打ってしまいやすくなったり、影響のある間接的な原因(課題からは遠い関係性にあるが影響力の高い原因)を十分に分析できていなかったり、組織間を跨ったプロセスの課題を見落としてしまったりするからである。

4.計画した変更内容

我々はこのような問題を解決する方法として、SPI にシステム思考アプローチの考え方を導入することを提案する。システム思考アプローチとは、システムダイナミクスから生まれた思考法で、課題を俯瞰的、長期的、多様な視点から捉え、全体像を様々な要素の繋がりと相互作用からシステムの構造として表し、最も効果的な解決策を考える方法のことである。システム思考アプローチを導入することで、ロジカル・シンキングの分析ツールで難しかった全体最適化や本質的な問題解決を図っていくことが期待できる。

システム思考アプローチでは、洗い出した課題の原因分析をする際に、その課題に関連する変化のパターンから因果関係を描き出していく。描き出した図はループ図と呼ばれ、増減や大小、長短など定量的な変化のある状況を変数とし、変数同士の因果のパターンをループ図として表す。変数の例としては、プロジェクト数、人員、品質、売上、生産性、評価など、増減のある要素であれば変数となる。単に因果関係を描き表わ

すだけでなく、定量的な変化のパターンの因果関係をシステムの構造として表している点が、ロジカル・シンキングの分析ツールとの違いである。定量的な変化の因果を描き表すことができるため、改善時の成果効果がより把握しやすくなる。

5. 変更の実現方法

我々は SPI 活動において、Plan-Do-See(PDS)の流れにシステム思考アプローチの考え方を当てはめて実践することを提案する。まず、Plan フェーズの診断では改善したいテーマの設定と、課題の分析としてループ図を作成する。改善したいテーマは課題が相当するが、詳細に洗い出す必要はなく、「いかに品質を向上させるか」や「なぜ品質が低下するのか」というような単純なテーマでよい。設定した改善テーマに関係する定量的な変化のある状況を変数とし、思いつく限り変数を洗い出し、洗い出した変数からループ図を作成する。例えば、不具合数、試験項目数、試験時間、プロジェクト数などを変数として洗い出し、「プロジェクト数が増えると、1PJ あたりにかけられる試験時間が短くなる」「1PJ あたりにかける試験時間が短くなると、実施できる試験項目数が減る」「試験項目数が少ないと、不具合数が増える」など、それぞれの変数の因果を描き出す。特に、組織間を跨ったプロセスに着目して変数を洗い出すと、より本質的な問題構造を見出すことができ、真の課題や具体的な課題を確認することができる。また、ループ図の書き方の特徴として、起きている現象が好循環あるいは悪循環の方向に強まるのか、均衡が取れている循環なのかが一目でわかるようになっている。

次に Do フェーズの改善策の検討では、作成したループ図を用いて改善策を検討していく。ループ図にはシステム原型と呼ばれる典型的な問題構造のパターンが 7 種類あり、作成したループ図がこの 7 種類のパターンに当てはまるかどうかを検証することによって、改善策のヒントを見つけやすい。また、レバレッジ・ポイントを見つけることで効果的な改善が期待できる。レバレッジ・ポイントとは、小さな力でも大きな変化を起こせるような効果的な介入点のことである。レバレッジ・ポイントの見つけ方として、他の変数と多く繋がっているところは他への影響力が高いことから、そのような変数に着目したり、「結果的な数字」「物理的な構造の変化」「制度や組織上の構造の変化」「メンタルモデル」などレバレッジ・ポイントを考える 6 つの観点で見つけたりする。このようにシステム原型の検証やレバレッジ・ポイントを見つけることで、改善策の検討が可能となる。

最後に See フェーズの改善成果の評価では、作成したループ図に変化があったのか、変数に変化があったのかを検証することで効果を把握できる。

6. 変更後の状態や改善効果

現在、システム思考アプローチを用いた SPI を適用中であり、具体的な効果については検証しているところである。ループ図を描く前は課題が複雑に絡み合っており、どこから手を付けるべきか不明確なところがあり、関係者も真の課題が見えていないところがあった。しかし、ループ図を作成することにより、適用組織で起こっている課題の構造が明確になり、関係者に改善の意識を持ってもらうこともできた。特に、調達品質の問題は組織間を跨って発生していることが多く、CMMI での供給者合意管理プロセスの適用による診断だけでは、改善策を打っても十分な効果が得られないことがあり、システム思考アプローチの考えを導入することによって解決に導きやすいことがわかった。

7.改善活動の妥当性確認

ループ図の変数には定量的に変化のあるものを使うため、ループ図に用いた変数を計測することで定量的な成果を把握することが可能になると考えられる。また、作成したループ図からレバレッジ・ポイントを見つけることができれば、効果的な改善策を実行することが可能となる。しかし、最も効果的なレバレッジ・ポイントがどこにあるのかを見つけることが困難であり、より具体的な事例が十分に整理されていないことも見つけ出すのが困難な要因の一つである。また、改善策を実行してみて実際に効果があったかどうかを検証しないことには、それがレバレッジ・ポイントだったかどうか不明な点も、見つけにくい原因である。今後の課題としては、SPI に特化したレバレッジ・ポイントの見つけ方のノウハウや具体的な事例を整理していく必要がある。

1B2「多様な成熟度の組織をカバーする改善支援サービスの標準化の取り組み」中村淳(NTT データ)

<タイトル>

多様な成熟度の組織をカバーする改善支援サービスの標準化の取り組み

<発表者>

氏名(ふりがな)：中村 淳(なかむら じゅん)

所属：(株)NTT データ 品質保証部 品質マネジメント担当

<共同執筆者>

氏名(ふりがな)：渡辺 清孝(わたなべ きよたか)

所属：(株)NTT データ 監査部

氏名(ふりがな)：大鶴 英佑(おおつる えいすけ)

所属：(株)NTT データ 品質保証部 品質マネジメント担当

<要旨>

弊社では長らく、CMMIモデルに基づく改善に取り組んでおり、3組織でレベル5を、1組織でレベル4を達成している。

しかし、グループ内にはCMMIモデルに基づく改善に取り組むことが費用面、体制面等で困難な組織も存在していることから、そのような組織を対象として、その組織の現状に応じて、適切な支援を適切なタイミングで実施する改善支援サービスに取り組んできた。

しかし、費用面、体制面等で制約の大きい組織に対して適切な支援を行うために支援内容はオーダーメイドで決定していたことから、サービスを提供する全社SEPG側の要員不足等が問題になっている。

その問題を解決するために、今まで実施してきた支援内容を「IDEALモデル」に紐づけて体系化することにより、改善支援サービスの標準化を進めることで、本サービスの対応可能な要員を増やし、かつ、支援活動の立ち上がりを早くすることで、より多くの組織を支援できる体制を構築することを狙った。

<キーワード>

- ・SEPG側作業の標準化
- ・IDEALモデル
- ・CMMI SVC

<想定する聴衆>

- ・SEPG、品質保証等、主として他組織の支援に取り組んでおられる方々

<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

NTTデータグループには全社の標準となる開発プロセスが存在しているが、顧客からのコスト削減・納期短縮要望、要望されるサービスレベルの違い、各組織の歴史・風土の違い、提供するサービスの多様化等の要因から、各組織が提供するサービスの品質に大きな差が生じており、中には、全社の利益を大きく圧迫するような問題プロジェクト化してしまっているものも散見される。

このような改善に取り組む必要のある組織に対し、長年 CMMI モデルに基づく改善支援に取り組んでおり、3 組織でレベル 5、1 組織でレベル 4 を達成してきたが、その反面、品質の低いサービスを提供してしまう組織は予算、体制等の制約が大きいことから、CMMI モデルに基づく改善に取り組もうとしても、費用、要員のスキル（モデルを理解した要員の確保が難しい）等の問題から途中で挫折してしまう例が非常に多かった。

そのため、このような CMMI モデルに基づく改善に取り組むことが難しい組織に対し、当該組織の提供するサービス品質が向上するまで、適切な支援を適切なタイミングで提供することを目的とした新たな改善支援サービスを立ち上げた。

ただし、このサービスは、「その組織の現状に応じて、適切な支援を適切なタイミングで実施する」ものである。具体的には、改善に取り組む必要性を現場メンバーに腹落ちさせた上で実行する動機付け、危機感を共有し、標準類を上から落とすのではなく、現場の仕事のやり方をベースに、その会社、組織にあったプロセスと一緒に構築している。そのため、支援内容は対象組織毎にオーダーメイドで提供する必要がある。

2. 改善前の状態／改善前の状態をもたらした原因

支援内容をオーダーメイドで決定する際の手順は、以下の通りとなっている。

- ① 組織の現状をヒアリング等で把握し、問題点を明確化する。
- ② 明確化された問題点を改善するために、どのような支援を行うかを検討する。
- ③ 検討結果をもとに、実施する支援内容を対象組織側と調整する。
- ④ 対象組織側と合意した支援内容について具体的な実施スケジュールを定め、支援を実施する。
- ⑤ 支援内容について追加フォローが必要な場合、追加フォローを実施する。

この手順で実施することにより、対象組織に対するきめ細やかな支援を実施することは可能であったが、1 組織への支援にかかる工数が大きくそれに対して、本サービスへの対応が可能な要員が非常に少なかったことから、支援を希望する全ての組織を対象として支援を行おうとしても要員が不足してしまい、対象組織数を増やすためには対象組織に対する支援内容が薄くせざるを得ないというジレンマに陥っている。

3. 計画した変更内容／変更の実現方法

コストセンターである我々全社 SEPG は、要員を大きく増やすことは困難である。そのため、支援対象組織数を増やすためには、既存メンバーの中で、本サービスへの対応が可能な要員を増やすことによって対処することが現実的である。

しかし、本サービスは「各組織の現状に応じて、適切な支援を適切なタイミングで実施する」ものであることから、プロセス、モデル等について幅広い知識が必要となるため、対応が可能な要員を増やすことは教育等様々な課題に対処する必要があり、困難である。

そのため、今まで本サービスで各組織に実施してきた支援内容を、プロセス改善活動を計画・実行する際の一般的

なモデルである「IDEAL モデル」に紐づけて体系化して、フェーズ毎のベストプラクティスを選定し、かつ、テーリングの基準（およびテーリングの例）を明確化することにより、「各組織の現状に応じて、適切な支援を適切なタイミングで実施する」サービスを標準化する活動に取り組むことにした。

なお、この契機に CMMI SVC を参照することにより、本サービスを継続的に改善させる仕組みを取り込むことにも取り組んでいく。

4. 変更後の状態や改善効果／改善活動の妥当性確認

現在は活動を開始したばかりであるため、具体的な効果はまだ出ていない。

ただし、本活動が完了した暁には、サービスが標準化されることにより、「より多くのメンバーが各組織の支援活動に携われる」効果や、「活動の立ち上がりが早くなるため、1 組織の改善サイクルが早くなる（ひいては、次の組織の支援活動に移ることが早くなる）」効果等が得られると考えているため、本活動を継続し、効果を測定していく予定である。

また、以前から実施している CMMI モデルに基づく改善支援活動との統合も狙っていくことを検討している。

以上

1B3「現場発ナレッジの全社共有の取組み」相澤武(インテック)

<タイトル>

現場発ナレッジの全社共有の取組み

<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 先端技術開発本部 技術部

<共同執筆者>

なし

<要旨>

事業部制組織の場合、事業部門毎に業務特性が異なるため、共通の標準プロセスを全社に一律に浸透させるのは難しい。また、事業部門毎に保有している有用なナレッジの共有化も同様である。双方ともに難易度をあげている理由の一つにテラリングがあると考えられる。

本発表では、事業部門毎に保有している有用なナレッジを全社で共有するために、テラリングの課題をどのように乗り越えたか、その取組みについて紹介する。

<キーワード>

QCD、ナレッジ共有、テラリング、プロセス改善、ワーキンググループ

<想定する聴衆>

SEPG 初心者

<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、お客さま満足や収益の確保などプロジェクトの成功率を高めるために「良いプロセスが最上の成果、高い生産性をもたらす」という考え方のもと全社の業務プロセスの標準化に取り組み、展開を図ってきた。事業部制組織をとっていることもあり、全社の標準を事業部門で利用する際には、事業部門の特性にあわせてテーラリングを行う必要がある。

しかし、各事業部門の規模や業務特性などにより、十分にテーラリングができず、標準が浸透していない部門もあった。

また、事業部門によっては、これまでの部門での経験を基に考案した有用なナレッジを保有しているところもあるが、このナレッジの横展開もなかなか進んでいなかった。

全社レベルで QCD 向上が求められている昨今においては、全社の標準の浸透度をあげることはもちろんのこと、各事業部門が保有している強みを有用なナレッジとして、それぞれが全社展開するとともに、展開されたノウハウの適用率を上げていく必要がある。

2.改善前の状態

当社では、全社の業務プロセス標準はあるが、その浸透度は事業部門によって温度差がある。また、事業部門毎に工夫して考案した有用なナレッジがあるが、そのナレッジの横展開、全社での共有はなかなか進んでいない。

全社の業務プロセス標準や現場で保有している有用なナレッジを、スピード感を持って全社に展開し浸透、共有化を図ることにより、全社的な QCD の向上につなげることができると考える。

3.改善前の状態をもたらした原因（因果関係）

当社では、事業部制組織をとっており、各事業部門においては、それぞれ異なる業務特性を持っている。このため各事業部門内のプロジェクトでは、原則として全社の業務プロセス標準を各事業部門の業務特性にあわせてテーラリングを行った事業部門標準を参照することになっている。しかし、全社の業務プロセス標準の浸透度合いを調査すると、実状は要員や時間的な制約から事業部門標準の制定ができていない部門もあり、全社の業務プロセス標準の有効活用があまり進んでいないことがわかった。

一方、事業部門においては、これまでの部門での経験を基に QCD 向上のために考案した開発標準をはじめとし、開発フレームワークや共通ライブラリなど各種有用なナレッジを保有している部門がある。このような有用なナレッジを全社的に共有する仕組み（掲示板への掲載と全社発表会での発表）は既にあったが、ナレッジの活用状況をみると、前述した全社の業務プロセス標準と同様の理由から有効活用が進んでいなかった。

4.計画した変更内容

各事業部門で保有する有用なナレッジを、他部門でも容易に活用できるようにするためには、以下の点が必要であると考えた。

①直ぐに利用できる状態となっていること

事業部門毎に業務特性があるので、自部門の業務特性にあうようにテーラリングが必要である。しかし、部門によっては、テーラリングのための工数が取れないところもあるので、全社に公開された時点でテーラリング不要の状態（自部門の業務特性にあったもの）になっている必要がある。

②納得感を持って利用できること

実際に使用するにあたっては、実績のある取組みが良い。実績のあるものであれば、良い点や改善が必要な点などがわかっているので、それらを踏まえたうえで納得感を持って利用することができる。

5.変更の実現方法

各事業部門で保有する有用なナレッジを、スピード感を持って全社に展開し浸透を図るために以下の施策を行った。

①有用なナレッジを発掘する仕組み

有用なナレッジの共有化を図るには、まず各事業部門でどのようなナレッジを保有しているのかを把握する必要があ

る。当社には以前より IKS（インテックナレッジシェアリング）という全社ナレッジを共有する仕組みがあったので、これを使用して、有用なナレッジの発掘を行った。

全社ナレッジとして採用するにあたっては、全社展開後に、ナレッジ提供元の事業部門に他部門からの問合せ対応等の負担がかからないように、全社展開後の問合せ対応等は、スタッフ部門で受けるような体制とした。

そのため、ナレッジの中でも、開発フレームワークや共通ライブラリなど、展開後にフォロー（使用方法がわからない、使用してみて動かなかった、環境の維持管理など）が必要なものについては、現状のスタッフ部門では体制が取れなかったため、今回は展開の対象外とした。

②発掘した有用なナレッジの共有化の方法（事例）

これまでに 2 パターンでの共有化を行った。共有化にあたって念頭においていた点は以下の点である。

- ・共有化の検討（WG や試行）には、主要事業部門の有識者に参画してもらう。
- ・成果物としてまとめあげるのは事務局が行い、事業部門からの参画者には作業負担はなるべくかけない。

今回の事例の概要は以下のとおりである。

（パターン 1）WG（ワーキンググループ）による汎用化（アプリケーション運用・保守プロセス標準の展開）

事業部門で保有していた「アプリケーション運用・保守プロセス標準」を、全社の主要事業部門から選出した有識者による WG の中で汎用化を行い、全社展開を図った事例。工夫点を以下にあげる。

- ・従来は、標準プロセスを作る際には、WG 形式で有識者が各自ベストプラクティスを持ち寄り、それをベースに作り上げていたが、今回の取り組みでは、既の実績のあるものをベースに検討を行ったので、できあがるまでの時間を従来よりも短縮することができた。
- ・WG で検討した結果は標準化ガイドとして整理し、プロセス標準とともに全社に公開した。また、標準化ガイドのエッセンスを e ラーニング教材化して公開した。
- ・標準化ガイドの中では、当社における保守サービスを代表的な 5 パターンに分類し、各パターン別にプロジェクトへの適用方法をガイドした。これにより、各事業部門での全社標準のテラリングの負担を軽減した。

（パターン 2）研修教材化（原因分析手法研修の実施）

事業部門で保有していた「原因分析手法」をもとに、研修教材を作成し、全社研修の一つとして展開を図った事例。工夫点を以下にあげる。

- ・ナレッジ提供元で作成していた説明会資料をもとに研修教材を作成した。
- ・作成した研修教材をもとに、主要事業部門の有識者参画のもと試行研修を実施し、参加者から出てきた意見をもとに研修教材及び研修カリキュラムのブラッシュアップを図った。
- ・研修の提供方法として、研修カリキュラムや日時を固定した集合形式ではなく、オンデマンド形式で、各事業部門の要望に応じて、開催日や研修カリキュラム等のカスタマイズも行った。

6. 変更後の状態や改善効果

今回事例であげた 2 つの取り組みについては、現時点で以下の実績があり、有用な現場ナレッジの全社への展開・浸透にはある程度効果があったと考えている。

①パターン 1）アプリケーション運用・保守プロセス標準の展開

- ・e ラーニングコース（2012 年 10 月開講）の受講完了者数は、2015 年 4 月末時点で 1397 名。

②パターン 2）原因分析手法研修の実施

- ・2014 年 10 月～2015 年 3 月までの間で、計 17 回、203 名（11 本部、3 センター、1 部所）に研修実施。

7. 改善活動の妥当性確認

従来は、各事業部門で保有する有用なナレッジが事業部門内だけの活用に留まってしまうことが多かったが、今回の取り組みにより、課題であった「各事業部門で保有する有用なナレッジをスピード感を持って全社に展開し浸透を図る」ための一つの成果が得られたと考える。

ただし、今回共有化の対象外とした開発フレームワークや共通ライブラリなどのナレッジについては、事業部門からのナレ

ジに対するニーズも高いこともあるので、今後は共有化の対象とできるように、スタッフ部門の体制を整備していく必要があると考える。また、今回紹介した事例では、事例を一つずつ実施していたが、今後は、よりスピード感を持つために、有用なナレッジについて、同時並行で進めていけるような仕組みの検討も必要である。

今後も今回の事例を参考にして、現場発ナレッジの全社共有化を推進し、全社での QCD 向上を図っていきたい。

以上

1B4 「プロセス改善は隗より始めよ～品質保証部の業務改善～」福本剛(NTT データ)

<タイトル>

プロセス改善は隗より始めよ ～品質保証部の業務改善～

<サブタイトル>

<発表者>

氏名（ふりがな）：福本 剛（ふくもと たけし）

所属：NTT データ

<共同執筆者>

氏名（ふりがな）：矢部 智（やべ さとし）

所属：NTT データ

氏名（ふりがな）：佐藤 慎一（さとう しんいち）

所属：NTT データ

<要旨>

弊社品質保証部では、これまで社内の品質マネジメントやプロセス改善に関する多くの施策を展開してきたが、ひとつひとつの活動を積み上げて実施した結果、定常作業が肥大化していた。そのため、上層部より新たな品質保証に関するミッションが提示された場合にも、それらの定常作業の遂行に労力を要し、リソースの観点から新たなミッションに十分取り組めない状況も発生した。そこで、品質保証部自身、のプロセス改善を実施し、定常作業の工数を大幅に削減した。

<キーワード>

品質保証部、作業効率化、モデル化、工数削減、手順改善、手順書、テンプレート

<想定する聴衆>

品質保証部門の方、（業務を問わず）部内全体の業務改善をしたい方

<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、これまでお客様に高品質なシステムを提供するため社内網羅的な品質管理能力の向上に取り組み、品質保証部が核となり社内に各種施策を立案展開してきた。これらの活動より、社内全体の品質管理能力のベースは定着し、今後は各現場が自律的に PDCA を廻す段階になってきた。

その結果、経営層からは、重点を絞った品質改善課題が示され、そちらの活動を実施する必要が出てきたが、一方で、品質保証部に対する業務効率化も併せて提示された。

2.改善前の状態

当時、品質保証部内には、社内への施策展開を実施するための定常作業が多く存在し、各部内メンバで受け持っていたが、これらの定常作業の遂行だけでチーム稼働の半分近くを費やしている状態であった。そのため、新たに経営層から提示された、課題に対して重点を絞った活動に注力するためのリソース手当に苦労していた。

3.改善前の状態をもたらした原因（因果関係）

チーム稼働の半分近くを費やしていた定常作業は、これまで品質保証部が社内へ品質向上の各種施策を展開するたびに、都度派生的に発生したものである。これらの定常作業は、どれも作業発生当初にアサインされた担当者がそのまま専担者となり遂行されたため、個々に属人化が進んでしまい、作業間の情報の流れや全体像が見えにくくなっていった。

各定常作業の専担者は、これまでの作業遂行の経験から、受け持ち作業について、手順など改善したい点等の認識は持っていたが、前述の理由で他の定常作業への影響が見えないため、改善に手を施せない状況であった。

また、彼らは他にも重要な作業も抱えていたため、受け持ちの定常作業については、とりあえず目の前にある業務を掃くことで精いっぱい状況で、チームメンバは定常作業については自発的な改善が起こりにくい状況であった。

4.計画した変更内容

定常作業の効率化計画は、以下のプロセスで検討し、作成した。

① 部内の全作業棚卸と実施対象範囲の決定

部内で遂行されている全作業の棚卸しを行った。棚卸された作業の中で、定常作業（一過性ではなく、定期的な間隔で繰り返し発生する作業）を抽出し、それらを効率化実施の対象範囲とした。

② 現状把握

現状把握として、効率化を施す対象範囲の全体像と各定常作業の関連を把握するため、全定常作業に対して、作業の発生源となっているデータを中心としたモデル化を実施した。作成後のモデルを個別と全体で見ることによって、作りっぱなしで誰にも使用されない成果物がないか、異なる作業で同一の成果物を作成していないか等、現状のデータ観点からの冗長や矛盾を踏まえた改善点を抽出した。

③ 効率化計画の作成

現状把握の結果をベースに、定常作業毎に効率化計画を作成した。効率化は、モデル化で明確になった改善点に着目して手順を改善すること等で実施することとした。

また、手順の中でアウトソース化が見込める作業についても検討した。

5. 変更の実現方法

効率化計画の実現は、作業を熟知している各定常作業の専担者に依頼するのが一番効果的である。しかし、彼らは皆、他にも重要な作業を抱えており、今回、効率化作業への協力を取り付けても、他の作業と並行状態となった場合は、（定常作業は、改善点はありつつも、現状のやり方でもとりあえず作業は廻るため）効率化作業の優先度は、常に先送りとなる恐れがあった。

そこで、効率化計画の実施対象メンバに対して、本作業を取り組み易くする環境を整えた。

① 業務課題の明示

「定常作業効率化」を部内の必須業務課題として明確に掲げ、それを受けて、実施対象メンバは、それぞれ受け持っている作業の効率化を正式な業務項目に位置付けることとした。

但し、先に述べたとおり、実施対象メンバ各人は、定常作業以外にも複数作業を抱えているため、本作業の開始・終了日は、チーム一律に設定することはせず、実施対象メンバ個別に、年間作業スケジュールを鑑み比較的稼働が空いている期間に目掛けて短期集中で計画設定し、実施管理も個別に行うこととした。

② 作業実施のための支援情報提供

実施者各人が作業効率化を、均一な品質、かつ決められた短期間で作業を完了できるよう、作業手順を定義するテンプレートと作業効率化を行うための手順（ポイント）を実施対象メンバに提供した。

（テンプレート）

作業手順を定義するテンプレートは、作業の末端単位（タスク）に対して、入力/出力、作業内容、担当者等を定義し、それらのタスクをフロー形式で可視的に記述できる様式である。

各定常作業の作業手順は、このテンプレートに反映することにより、いままで課題となっていた属人化が解消され、誰でも視覚的に認識できる形式に表現される。

（作業効率化の手順）

実施対象メンバに共有した、作業効率化の（主な）手順を以下に示す。

- ・定常作業の現行の作業手順の可視化を行う。（現行の手順を先に示したテンプレートに反映する。）
【※以降、可視化した現行手順をベースに】
- ・作業内で閉じ、かつ誰にも参照（利用）されない成果物は、タスクも含めて削除する。
- ・担当者間で、情報のやり取りが複数回発生している部分は、やり取りの回数を極力減らすよう手順を変更する。
- ・上記の手順改善に併せて、現行様式の改善（最適化）およびペーパーレス化を検討・実施する。

- ・（マクロやツール等で）自動化が可能なタスクがあれば、検討・実施する。

実施対象メンバに上記 2 点を提供したことで、メンバ各人の効率化に対するモチベーションが高まり、これまで各人が潜在的に抱えていた改善意識が効率化作業の結果として顕在化された。

本活動は、全体で 2 年の期間を要したものの、当初の効率化計画を概ね実施することができた。

6. 変更後の状態や改善効果

現状把握より、部内の定常作業は合計 20 個存在したが、それらに対する効率化の実施結果は以下のとおりである。

- ① 手順改善 & 手順書整備（属人化排除） : 10 作業
- ② (①+) 作業の委託化実施 : 4 作業

今回の活動を通して、各定常作業が可視化されたため、これまで課題となっていた属人化が解消し、定常作業の専担となっていたメンバを切り離すことができた。それによって、各作業のリソース割り当ては、その時の稼働状況に合わせて動的に行えるようになった。

また、各定常作業における、手順改善（無駄なタスクの排除等）と委託化により、部内における定常作業にかかる稼働が大幅に減少したため、改善前に経営層から提示された課題である、重点を絞った活動に注力するためのリソースを確保できるようになった。

併せて、部内メンバの意識改革として、上記の改善効果を実感できたことにより、従来からの定常作業に追われ続けていた閉塞感から解放され、メンバ全員が新たな経営課題を受けた活動への取り組みに邁進し始めることができた。

7. 改善活動の妥当性確認

今回の作業効率化による定量効果の試算は、年間 24 人日の工数削減であった。これらの削減工数は、経営層から提示された新たな課題検討の活動に振り分けることができた。

なお、今回効率化未着手であった 6 個の作業についても、今後更なる工数削減を図っていく。

1C1 「静的解析結果の見せる化」和良品文之丞(キヤノン I Tソリューションズ)

<タイトル>

静的解析結果の見せる化

<サブタイトル>

～並べて、比べて、区分けして、気づきを促す～

<発表者>

氏名(ふりがな)：和良品 文之丞 (わらしな ぶんのじょう)

所属：キヤノン I Tソリューションズ株式会社 品質監理本部

<要旨>

ツールによる静的解析結果を用いて、複雑で規模が大きい箇所をリファクタリングや重点的テストの対象とするよう啓蒙した。ところが現場からは解析結果の良し悪しがわからない、稼動しているものは変えられないなどの反応があり、具体的なアクションにつながりにくかった。

そこで、複雑度や規模の高い順に並べる、他の解析結果と比較する、しきい値で区分けするなど「見せる化」に努めた。その結果、CI 環境で定常的に劣化を確認する、しきい値をコーディングルールに反映してリスクを減らすなど、現場の改善活動につなげることができた。

<キーワード>

静的解析ツール、静的解析、MTM (Metrics Tree Map)、DSM (Dependency Structure Matrix)、コード行数、サイクロマティック複雑度、本質的 (エッセンシャル) 複雑度、保守難度、コーディングルール、しきい値、ゾーン分析、バブルチャート

<想定する聴衆>

品質保証の方、SEPG、静的解析ツールを導入している方、導入しようとしている方、ソフトウェアエンジニア

<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年から開始した品質改善委員会の活動の一環として、静的解析ツールの導入を決定し、ソフトウェアのプロダクト品質向上に向けて、ツールの共有、導入支援、解析サービスなどを実施した。特に、ツールの普及を図るため、現場に静的解析の実施を提案し、その解析結果を開発者に直接報告した。

報告の際には、メトリクス・ツリー・マップ（MTM）を用いて、規模が大きく複雑度が高いモジュールをモジュール名と数値を交えた具体的な報告や、ディペンデンス・ストラクチャ・マトリックス（DSM）を用いて依存関係を可視化し、変更による影響の大きい箇所を示すなどを行った。

それらを通じて、複雑度が高く、規模が大きいモジュールは、内容を理解することが困難で、かつ欠陥が入り込む可能性が高い、品質を脅かす箇所として、リファクタリングの対象とすること、重点的にテストを行うこと、変更の際にはツール等を活用して慎重に影響分析を行うことを提言し、啓蒙した。

2.改善前の状態

現場からは、静的解析の結果や品質のリスク指摘に対して、一定の評価が得られ、一部のプロジェクトではリファクタリングや重点テストへの取組みも行われた。その一方で、多くプロジェクトから、以下の意見が出された。

- どこが複雑で規模が大きいかは、開発者であれば改めて言われるまでもなくわかっている。
- 示された状況からは、それが良いのか、悪いのか、よくわからない。
- 今稼動しているものを変えるのは困難で、計画的にリファクタリングを組み入れるのであれば、判断基準が必要である。

その結果として、現場では何をしたら良いかわからず、またツールを活用するメリットも伝わらず、具体的なアクションにはつながっていなかった。

3.改善前の状態をもたらした原因（因果関係）

現場からの意見に対して、なぜそのような意見が出るのか、追加のヒアリングを行い、原因を分析した。その結果、報告に以下の問題があると思われた。

（1）情報量が乏しい

複雑度、規模に関しては、数値の高いものから1～2個を指し示したが、この個数では、ツールを用いなくても開発者が感覚的に判断できる範疇であり、ツールを使うメリットを訴求できていなかった。

（2）解析対象となったプログラム単体の報告

モジュール間の比較、コード行の種類の分類などを報告したが、その内容は、解析対象のプログラムで閉じており、ベンチマークのような評価ができていなかった。

（3）単なる事実の報告

静的解析結果を、わかりやすく図示したつもりであったが、値が示す意味や解釈に踏み込めていなかった。単に事実を提示しただけでは良し悪しはわからず、具体的なアクションにはつながらない。

4.計画した変更内容

前述の原因を踏まえて、現場の気づきを促すために、並べる、比べる、区分けする、という 3 つの「見せる化」を考案した。

(1) 並べる

MTM は、指定した範囲のモジュール間の相対値で順位を表示するため、最も複雑度が高いといってもその値は一意ではない。そこで、絶対値による順位を複数個並べて示すこととした。一つの画面に収まるように、並べる個数を 10～20 の範囲とした。

(2) 比べる

ツールベンダでは、代表的な OSS のプログラムの静的解析を行い、その数値をノウハウとして蓄積していると聞いた。そこで、規模の大きな OSS のプログラムを解析した結果と比較することとした。

(3) 区分けする

文献からメトリクスに対するいくつかの指標を得た。そこから複雑度と規模にそれぞれしきい値を設定し、四象限で表現したときに、双方がしきい値よりも大きなゾーンを、危険ゾーンとして示すこととした。

5.変更の実現方法

計画した「見せる化」の実現方法、具体的な報告内容は以下のとおりである。

(1) 並べる

当初は、規模の大きいものを 10～20 個選び降順に並べた棒グラフで示した。後に、文献から得られたしきい値を併用し、しきい値をグラフ上に示したり、しきい値を超えたモジュールを抽出して降順に並べたりして報告し、より説得力が増すように工夫した。

(2) 比べる

当社は代表的な OSS の分析結果と比較したが、ソフトウェアの特性が異なるなど、必ずしも有効な情報とは言えなかった。そこで、リファクタリングを行ったプロジェクトの協力を得て、リファクタリングの前後の比較を行い、リファクタリングの効果を示すことができた。また、バージョンアップの前後の比較、蓄積された過去の解析実績との比較を行った。

(3) 区分けする

当初はサイクロマティック複雑度 20 超えとコード行数 100 行以上で、危険ゾーンを示した。その後、サイクロマティック複雑度と本質的（エッセンシャル）複雑度から導出される「保守難度」に注目して、サイクロマティック複雑度と保守難度の危険ゾーンに、モジュールの規模をあらわすバブルチャートを重ねて示した。

6.変更後の状態や改善効果

結果報告を受けた現場の意見は、以下のとおりである。

- どこが複雑で規模が大きいかわかっているつもりでいたが、具体的な数値では掴めていなかった。また、しきい値を超えたものがどれだけあるのかは、わからなかった。
- 他の解析結果と比較できると、自分のプログラムがどの程度の作りなのか、考えさせられる。良いか悪いかではなく、値が突出しているものは気になる。
- 危険ゾーンはわかりやすい。またそのゾーンにどの程度の規模のものがいくつあるのか、明確である。このゾーンに入らないようにしたい。

その後、それぞれ自分の製品に合った改善活動が始まった。以下、三例を紹介する。

(1) 継続的な統合 (CI) 環境で劣化をチェック

CI 環境を持つプロジェクトでは、ビルド時に静的解析ツールを自動的に起動し、メトリクスを収集するようになった。ビルドの都度、メトリクスを用いてコードが劣化していないかを確認している。このプロジェクトでは夜間（深夜）にビルドを行う、かつ自動化されているため、効率的にライセンスを活用できている。

(2) ルール化による劣化の未然防止

歴史のある製品では、稼動しているコードを変更するリスクが高く、リファクタリングには消極的であった。このプロジェクトでは PM の発案で、複雑度と規模に上限を設け、コーディングルールとフェーズ移行の判定基準に追加した。新たに作成したり、修正したりするモジュールは、その値を超えないこと、どうしても超えた場合には、PM と一緒にその妥当性をレビューする、という活動を行っている。

(3) バージョン間での構造確認

比較的バージョンアップのサイクルが短い製品においては、バージョンごとに静的解析ツールでメトリクスを確認すると共に、構造解析を行い、アーキテクチャの健全化を図っている。

7.改善活動の妥当性確認

静的解析ツールの活用は、その後も継続している。ルール化したプロジェクトでは、設定した基準での運用を行っており、その取組みが品質改善委員会でも報告されている。

CI 環境やバージョンアップで劣化を確認しているプロジェクトでは、ツールの予約状況、解析依頼とバージョン発行時期、ツールのアクセス件数を通じて、定常的に実施されていることが確認できた。

特に、ゾーンは危険度を伝えるのに有効な表現方法である。また、しきい値や他のプログラムと比較することによって、現場が自ら考え、気づきを得て、現場に合った活用方法を自ら開拓したことが、活動の定着化のポイントである。

残存課題としては、引き続き解析結果を資産として蓄積し、タイプ別や類似のソフトウェアとの比較ができるようにすること、一部のベンダで取組みが始まった構成管理情報との組み合わせ分析、そして品質との相関を確認してより品質の向上に寄与することである。

<発表内容>

1.背景

今回の改善活動に取り組んだのは、対象プロジェクトとなった開発者と品質管理部員である。

主体である開発プロジェクトは組込みの派生開発をここ数年に渡って行っており、リリースを数ヶ月単位で繰り返している。問題は納期に間に合わず、再リリース日を設定することが多いことにある。当然、顧客との信頼関係や当社の利益にも大きな影響を与えてしまう問題である。

この問題に対し、上位組織としてはスケジュール厳守と利益確保、つまりはプロジェクトの正常化を求めている。

この要求を実現するために、現場ではいくつかの開発プロセスの改善を計画し、そのうちのひとつとして上流工程での品質の作り込みの実現を目標に掲げた。この目標に対し、品質管理部では ODC を用いて設計欠陥の真の原因を突きとめ、リリースを遅らせている品質問題を解決するため、改善活動を開始した。

2.改善前の状態

当社の某派生開発プロジェクトにおいて、テスト期間が長くなってしまふことにはいくつかの要因がある。

そのなかでも課題として挙げられるのは、欠陥検出がテスト後半になっても多く検出され、欠陥修正に計画以上の工数を取られ、さらに障害の影響に対して品質保証をコミットするには、更なるテスト期間の延長が必要とされてしまうことにある。

今回の対策のひとつとしては、テスト工程での欠陥を少なくするためにはどうすればよいか？から議論が始まり、上流工程での欠陥混入を防止する対策を立てることが抜本的であり、かつ長期的にみれば懸命な解決策であるとの意見に集約された。

そのためには品質の良い設計を行うことが要求されるが、それを実施するためには現状の課題を見えるようにしなければならない。そこで設計プロセスの調査と検証を実施し、設計方法やレビュー実施について調査を進めているうちに、開発者は検出された欠陥に関しては、修正期日に追われ、とにかく不具合を起こさないように対処しているだけのことが多いことに気付いていった。

果たして欠陥を顕在化させたものは何なのか？このまま設計者が真の原因に気が付かなければ、同様な欠陥をまた作りこんでしまうことになる。また、それらはチームの設計者たちとの気付きやノウハウにもなっておらず、疲弊する開発が続くだけである。

真の原因を突きとめることには時間がかかるが、得られる効果も大きいはずだと意見に後押しされ、ODC による分析作業を始めた。

これらの分析活動から開発プロセスの改善策として、欠陥を埋め込む原因を突き止めることにより、対策を講じるための有効なフィードバックが実施でき、設計品質の向上を図れると考えた。

3.改善前の状態をもたらした原因（因果関係）

設計欠陥の真の原因を突き止め、しかるべき対処を取れば、テスト工程の欠陥数が減るはずだと判断した。

その根拠としては、テスト工程で発見された欠陥がどの工程で作られたか、その混入工程を調べ、その欠陥が設計でミスをした欠陥か、コーディングでミスした欠陥かに分類し、それらの計測値を比較した。

すると、設計型欠陥数の割合が主だった混入工程で 80%を超える数値を示した。

これらの調査から設計欠陥の特定が重要だと判断され、次に、どうやって原因を特定すればよいか、議論が移っていった。

4. 計画した変更内容

改善前は、テスト後に評価メンバーが欠陥分析にあっていた。評価者によって、その分析結果の粒度やアウトプット形式がマチマチであり、どうしても分析能力が高い評価者に分析業務が集中してしまい、ボトルネックとなってしまうことが多々あった。分析能力が高い評価者を育てることも重要だが、誰が分析しても同じ結果になるような方法がないかを模索し、不具合分析手法のひとつ、ODC 分析に行き着いた。

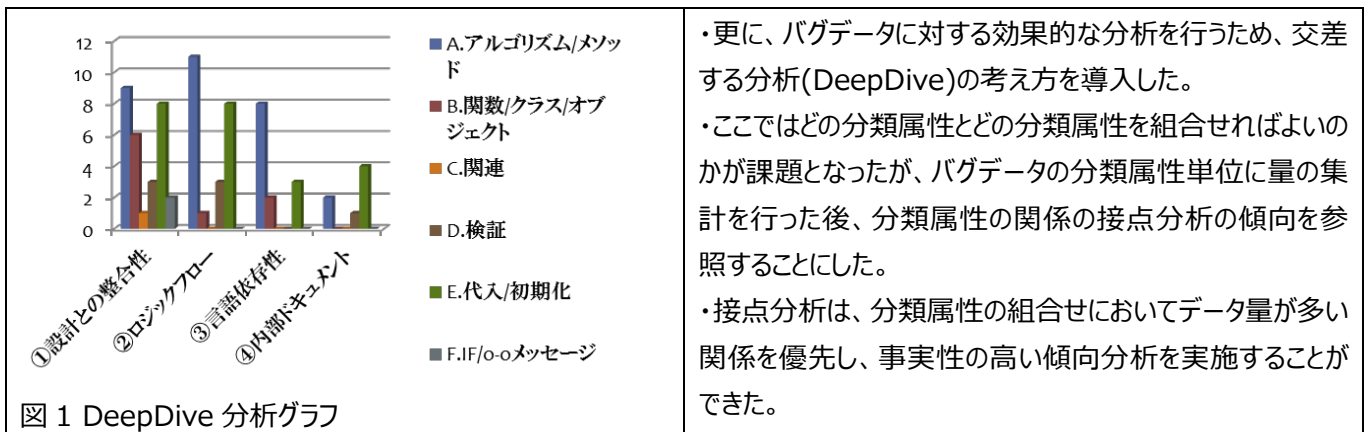
まずここでテスト検出された欠陥について、ODC 分析を適用し、一定の評価と改善を得ることができた。

同様に設計者にもレビュー時に欠陥を分類することを始めた。ODC 手法に則り、レビュー議事録をもとに、設計に特化した分類属性を各欠陥に対して項目設定し、その分類属性毎の集計をグラフ化した。この分類集計することによる量と質の傾向から、プロジェクトの陥りやすい欠陥の傾向を見ることも可能になってきた。これにより、改善案もその結果を検証した上で、順次進めていく道筋をもてるようになった。

5. 変更の実現方法

ただ、原因系の ODC 項目設定は設計者によって、キーワードがマチマチになってしまう課題は簡単に解決できず、設計側で必要と思われるアウトプットとしては精度が低いのではないかという疑念が残ってしまった。そこで欠陥の分類属性設定の正確度を上げるために、分析の手順を論理的に組み立て、属人的にならない方法を検討した。

属人的な項目設定にしないためには、設計者にとってわかりやすい分類属性項目を作り、どの項目を選択すればいいか、欠陥を選別するフローチャート的な手順書を作成し、設計者への説明も適宜実施した。



またアウトプット形式もフォーマットを作成することにより、データのグラフ化もしやすくなり、同じバグ項目であれば、設計者による偏りが減り、同じような分析結果が残せるようになってきた。

6. 変更後の状態や改善効果

ODC 分析の結果の妥当性を検証するために、設計欠陥の真の原因を突き詰める活動プロセスも導入した。

これには、なぜなに分析を用いて深掘を実施している。

この活動結果から、間違った設計判断をしたり、ある制約条件を忘れていたり、要求を誤解していたりといった原因とその理由といった定性的な分析、そして欠陥の種類が量的にも見えるようになってきた。

この分析中で真の原因として多く結びついたのが、設計表現の不備である。

実際の設計においては不明確で、意味が幾通りにも解釈できるような設計記述が多く見られ、それをノウハウとか暗黙知とか開き直す設計者が多いこともわかってきた。

要するに、設計者が要件解釈のうえで妥協したり、十分な情報なしに設計判断したりすると、すべてのテスト工程にはより多くの欠陥が混入し、残存してしまう。

逆に、明確で完全な設計表現は、作り込みの欠陥を削減するのに非常に役立つということだが、これを改善するのは指

摘するだけでは治らず、非常に難しいということがわかってきた。

一般的には、論理的な文章を書く、図的表現を使用する、擬似コードを記載するなどといった理解しやすく、勘違いの幅を狭くできる表現を磨いていくことが必要であり、そう簡単に改善できるものではない、ということである。

そこで長期的な改善活動を計画し、設計改善の指導を開発部門でリードしていくことを決めていった。試行錯誤を行っている状況ではあるが、現在は複雑な設計を記述する場合には、シーケンス図や内部ブロック図といった図的表現を必ず用いて説明しているかをチェックし、レビュー時に確認指示するやり方で周知を図っている。

同様に、これまで見逃してきた文章表現についても、論理的でない文章は不明なまま残さず、修正を必須とすることをレビュー時に指摘しており、徐々に改善が見られる設計書は増えてきている。

また、評価者では以前より、仕様書から要求仕様が読み取れない、わからないといった声が上がっていたが、こうした改善活動後はテスト設計書が作りやすくなった、という定性的な効果も挙げられている。

こうした改善方法で設計者が念入りな作業を行うことにより、効果はプロセス全体に現れてきた。ここ数件のプロジェクトの測定値では、レビュー工程とテスト工程での欠陥検出率が逆転し、テスト工程で欠陥数が少なくなってきたプロジェクトも出てきた。

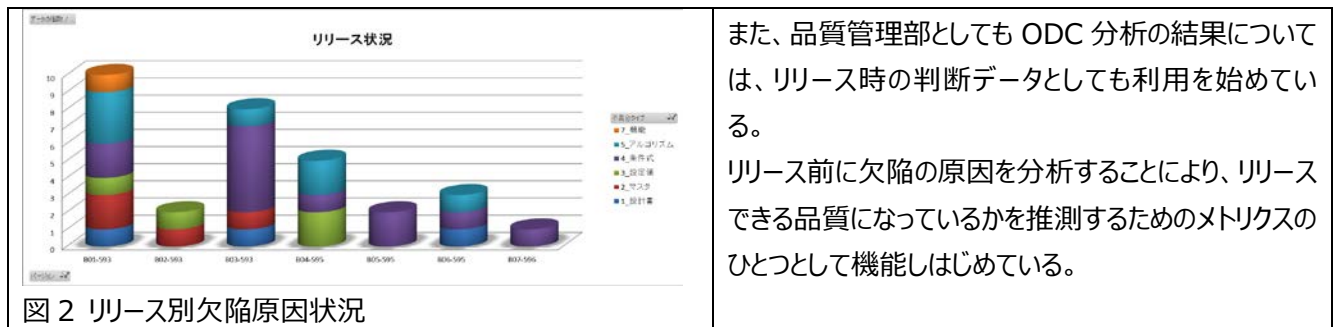
まだすべてのプロジェクトが同様の傾向を示しているわけではなく、リーダーの特性によって異なっていることもわかってきている。リーダーの方法論を変えることは難しいが、数値の浸透とともに変わっていくことを期待している。

7.改善活動の妥当性確認

当初の目的である、リリース日の変更が必要なプロジェクトはまだ無くなったわけではないが、改善傾向に向かっていることがプロジェクトサマリから見るできるようになってきた。

またテストでは重度の欠陥検出は少なくなり、欠陥数そのものが減少したことにより、欠陥の平均除去コストが改善されているプロジェクトも出てきた。欠陥除去コストは、ドキュメント修正 + モジュール改修 + 回帰テストの工数を計測したものであるが、加えて評価者からの修正依頼・返却期間が短縮され、テスト期間の正常化につながったと思われるケースも見られている。

もちろん ODC 分析やなぜなに分析の成果だけではなく、開発部門やチームの改善意識によるものではあるが、改善成果の一助になっているとの現場からの声も聞いている。



今回は、問題のあった派生開発プロジェクトに合わせて改善活動を実施してきた結果を報告している。

今後はこの改善成果を違う開発、別なプロジェクトに適用していけるかを、次のステップとしている。

1C3 「メトリクスでソースコードの美しさは判断出来るのか？」 山口祐史(富士フィルムソフトウェア)

<タイトル>

メトリクスでソースコードの美しさは判断出来るのか？

<サブタイトル>

内部品質に着目した品質管理の試み-その現状と今後-

<発表者>

氏名（ふりがな）： 山口 祐史（やまぐち ゆうじ）

所属：富士フィルムソフトウェア 基盤技術グループ

<共同執筆者>

氏名（ふりがな）：

所属：

<要旨>

ソースコードの品質は、バグの多い少ないと言う外部品質ではなく、保守容易性、拡張性、再利用性といった内部品質、つまりソースコードの美しさによってでなくては判断できない、というのが弊社の結論でした。しかしそれは同時に、どのようなメトリクスで測定出来るのか、測定した数値はどのような閾値で判断出来るのか、その閾値はどのように制御・維持・管理できるのか、といった問題に向き合う事でもあります。本発表では弊社におけるこの問題に対する取り組みをご紹介します、ご意見を伺って今後に役立てたいと思っています。

<キーワード>

メトリクス、品質管理、静的解析、コードクローン、CI、ALM、複雑度、リファクタリング、検収条件、ビッグデータ

<想定する聴衆>

品質保証関係者、プロジェクトリーダー、プログラマー、ソフトウェアエンジニア

<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(富士フィルムソフトウェア)では、バグの多さと言う外部品質の低さに悩ませられ続けて来ました。そのためCMMIによってプロセスを整備し、その上でソースコードの品質を保守容易性、拡張性、再利用性といった内部品質、つまりソースコードの美しさに着目し、如何にそれを維持管理できるかに取り組んできました。

その結果、100ステップ以上の巨大関数、30行以上のクローン関数、は全てゼロ件(新規、改造部分)という品質管理目標を、パイロット試行を経て今期初頭より全プロジェクト、全協力会社において品質管理計画に盛り込む、という規定を導入し運用を開始しています。

また、それを実現するためのツールをCQM (Code Quality Management) として、全実装担当者のローカルでの実施環境を「パーソナル CQM」として、またナイトリーでのコミット状態でのプロジェクト全体での遵守状況のレポートシステムを「チーム CQM」として提供しています。

2.改善前の状態

しかし、こうした品質管理目標は一律に終了・検収基準とすることは出来ません。ある許容範囲の閾値を合理的に定められない限り単なる努力目標に終わってしまいます。よって、合理的な閾値、およびそれを適切に表すことのできる指標が必要となっていました。

一方、こうした閾値を検討する上では、弊社内の過去事例からある程度の数値は蓄積されていました。しかし、それを合理的に説明する根拠はありませんでした。

3.改善前の状態をもたらした原因 (因果関係)

弊社内の過去事例から定量的なバグの多さと言う外部品質の低さと、100ステップ以上の巨大関数、30行以上のクローン関数、との相関は、明らかでした。また、これらに付随するメトリクス収集も行っていました。

しかし、こうしたメトリクスを品質判断の閾値として使用するに対しては、ごく一般的な指標を直接的に当て嵌めるだけで、製品全体の品質判断の閾値としては設定していませんでした。例えば、複雑度は10以下であるべきか、CBOは5を超えてはならない、という一般的な閾値はありました。しかし、それらをどれくらい違反したら本当に保守容易性、拡張性、再利用性といった内部品質は許容できなくなるのか。それは本当に複雑度 ≤ 10 やCBO ≤ 5 なのか、20や50ではないのか。100Kの製品にどれくらいの個数の割合まで許されるのか。個数ではなく規模比率ではないのか。いやワーストの絶対値ではないのか。こうした議論は突き詰めては検討されていませんでした。

更に、仮にこうした指標と閾値が設定出来たとして、それは制御可能なものなのか、といったことも突っ込んで議論はされていませんでした。例えば、仮に複雑度が10を超える関数は、製品全体の関数数の10%を超えてはならない、と言う閾値が設定出来たとして、これは制御可能か、と言う問題です。100K超の製造を猛然としているプロジェクトに対して、どのサブシステムのどのクラスのどの関数の複雑度をどのように下げれば、この閾値を維持管理できるかは、ほぼ制御不能です。一律に開発環境で複雑度10超のソースはコミット不能とでもしたら可能かもしれませんが、それが無意味なことは自明です。

4.計画した変更内容

そこで制御可能かどうかはともかく、内部品質を表すメトリクスとその閾値を突き詰めて検討し、弊社で蓄積した過去事例で納得の行くメトリクスとその閾値を検討しました。

5. 変更の実現方法

弊社で蓄積した過去事例で納得の行くメトリクスとその閾値を割り出すために、実績のある OSS のメトリクスと比較することで、合理的な根拠とすることを試行しました。

具体的には弊社の過去事例（13 例）と実績のある OSS（9 例）を、個数比率、規模比率と言った二次的な導出指標も含め 266 のメトリクスで集計しました。それと同時に弊社での主観的な評価、つまりソースコードの美しさを目視のソースレビューにて 5 段階評価を実施しました。それぞれを別々に統計分析し、5 段階評価とその他のメトリクスの相関分析を実施し、その上で、弊社の過去事例での 5 段階評価の相関データの集合と OSS での 5 段階評価の相関データの集合を突合せ分析を実施しました。その結果、弊社の過去事例と OSS ではいくつかのメトリクスで同種の傾向が見られることが分かりました。つまり、結果的に主観的な 5 段階評価に相関するメトリクスは、OSS でも同等に相関するため、弊社の過去事例での数値が合理的であることの根拠を得ることが出来ました。その結果、「メソッドステップ数 100 行超個数比率」（メソッドの実行行数が 100 行を超えるメソッドが全体のメソッド数に占める個数の割合）や「メソッドクローン個数比率」（メソッド単位のクローン数が全体のメソッド数に占める割合）といった指標が選定され、またその閾値も設定する事が出来ました。発表では詳しい指標と数値をご紹介します。

6. 変更後の状態や改善効果

合理的な根拠を得ることが出来たので、品質管理目標を具体的な終了・検収基準として明確に打ち出すことが可能になりました。

ただ、この分析の過程で今回のやり方の限界があることもはっきりしました。それは製品全体の 5 段階の主観評価との相関分析です。このやり方では主観の粒度が粗過ぎ、細かな部分の内部品質の分析には限界があります。今後はこの点の改善に努め、より細かな粒度での主観評価データの収集プロセス(メソッド、クラス単位での内部品質データの収集)を立ち上げ、具体的な分析に役立てる予定です。発表では、その点を「ソースコードはビッグデータである」との観点からご紹介いたします。

7. 改善活動の妥当性確認

最終的には失敗プロジェクト数の減少、失敗コストの減少、といった費用対効果を証明していくこととなりますが、取り組みは始まったばかりですので、今後約 3 年を掛けて追跡していく計画で進めています。

また、同様な取り組みの他社の事例なども参考にさせて頂き、妥当性を検討していきたいと考えていますので、よろしくお願ひします。

1C4 「コンカレント開発に向けたアーキテクトチーム導入による開発プロセスの改善」 林健吾(デンソー)

<タイトル>

コンカレント開発に向けたアーキテクトチーム導入による開発プロセスの改善

<サブタイトル>

車載ソフトウェア開発における進化的プロトタイピングの導入

<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>

アーキテクトチームを組織することによる、車載システムの新規開発における継続的なコンカレント開発方法の運用を提案する。短期開発と内部品質の確保を両立しながら、車両の各納入フェーズに遅れることなくソフトウェアをデリバリーし続けることを狙う。研究開発部門と製品開発部門がコンカレントエンジニアリングのアプローチで、互いの開発アクティビティを整合しながら製品を進化させる。アクティビティの中心にアーキテクトチームを置くことで、整合と継続を実現した。その効果を示すことで開発方法の有効性を示す。

<キーワード>

車載システム開発、品質確保、短期開発、プロトタイプ開発、使い捨て型プロトタイピング、進化的プロトタイピング、コンカレントエンジニアリング、フィードバック、アーキテクトチーム、継続的デリバリー

<想定する聴衆>

ソフトウェアエンジニア、組込ソフト開発者、SEPG、アーキテクト、プロジェクトマネージャ

<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

車両周囲の環境をセンシングして車両制御する車載システムの新製品を開発している。従来開発では、研究開発部門が実車両上にプロトタイプ開発する仕様開発フェーズと、製品開発部門が仕様を入力にして製品ソフトウェアを開発する量産開発フェーズの、2 フェーズ開発方法を採用している。近年、システムの高度化、複雑化が進む一方、車両メーカーからは商品力向上のために高品質かつ短期間での開発が求められるようになった。

2.改善前の状態

量産開発フェーズの開発期間が短縮された結果、使い捨て型プロトタイピングに基づいた2 フェーズ開発方法から進化的プロトタイピングへの転換が必要となった。しかし、仕様開発フェーズで開発されたプロトタイプでは製品としての内部品質の確保が充分でなく、製品展開の派生開発において品質リスク、納期リスクが生じ得る。

3.改善前の状態をもたらした原因（因果関係）

仕様開発フェーズでは、顧客と協調してフィーチャと技術を素早く開発することを優先し、内部品質の確保をスコープとしていない。例えば、保守性の指標である複雑度を測定したところ、開発規模に比例して複雑度は大きくなり、品質としての低下がみられた。内部品質を確保する技術は製品開発部門が保有しているが、開発期間の制約から、開発されたプロトタイプを引き継いだ上で確保できる品質レベルは限られる。

4.計画した変更内容

内部品質の確保課題を解決しながら進化的プロトタイピングによる開発期間の短縮効果を維持することを狙う。そのために、コンカレントエンジニアリングのアプローチにより、仕様開発フェーズから製品開発部門が参画し、協働して製品開発する方法を提案する。開発人員が増えることで生じる現場の混乱を軽減するために、アーキテクトを両部門の中間に置き、内部品質の確保責務と両部門のアクティビティをすり合せて整合させる役割を与える。各部門は協働しながらもアーキテクトを媒介することで、互いに疎結合を保てるようにする。この仕組みでは新規開発における一回分の試作納入には耐えられるが、連続した納入では製品開発部門の協力が期待できない。そこで、アーキテクトをチーム化し、納入時の継ぎ目を補完するバッファとして機能させることで連続した納入へ対処する。

5.変更の実現方法

超音波センサシステムの新製品開発において、2つの納入フェーズに跨って提案方法を導入した。アーキテクトチームは研究開発部門内に組織した。各チームの基本アクティビティは従来の開発とは大きく変更しない。アーキテクトチームの基本アクティビティは内部品質を確保するための設計を受け持つ。納入時は、製品開発部門が担っていた検証／検査を担う。アーキテクトにはアクティビティを整合するための2つの重要な役割を持たせた。Coplienの組織パターンで提唱される「門番」と「防火壁」のロールである。アーキテクトは製品開発部門に対して門番として機能する。プロトタイプ開発のプロダクト情報をリアルタイムに伝えることで、製品開発部門で生じ得るムダを軽減する。また、アーキテクトは研究開発部門に対して防火壁として機能する。製品開発部門から研究開発部門のメンバ／プロトタイプへのアクセス／影響を最低限に抑えることで、研究開発部門のアクティビティを妨げず、開発速度が損なわれないようにした。

6. 変更後の状態や改善効果

開発期間短縮効果として、1 回目の納入に掛かる開発期間を測定した。従来の 2 フェーズ開発方法に比べて、全体期間として 41.7%の開発期間短縮効果が得られた。内部品質の確保効果として、処理時間、メモリ使用量、複雑度を監視計測した。2 つの納入フェーズを通して、各測定値が継続的に改善かつ維持され、製品としての内部品質確保を果たすことができた。また、組織間のアクティビティの整合として、プロトタイプ開発に並行して実施した製品開発部門による単体テストについて、95%の関数がやり直し 1 回までで完了したことが確認できた。内部品質確保のためにアーキテクトチームがアーキテクチャを把握することで、仕様開発フェーズのプロトタイプ開発においても処理時間やメモリ使用量へのソフト変更の影響が明確になり、研究開発部門でも内部品質を意識したアクティビティが誘導され、自発的に内部品質の高いソフト開発が促されるようになった。

7. 改善活動の妥当性確認

研究と量産、開発と保守のように異なる組織において異なる技術を統合しながらスループットを向上する方法として有効であるといえる。組織をひとつにまとめ上げることが可能なプロジェクトであれば、Scrum のように開発を完結させることが望ましい。また、開発ドメインの技術が確立されていればモデルベース開発が望ましい。これからの製造業における組み込み開発のように技術が未確立で組織も容易に統合できない場合に有効な手段であると言える。但し、組織間を調停する能力と大きな負荷を乗り越える資質がアーキテクトチームのリーダーに求められ、多かれ少なかれ人に依存した施策となる。これらのアクティビティを詳細化して補助する技術を開発し、属人性を小さくすることが今後の課題である。

2A1 「通信ソフトウェア開発「短期リリース」への道」 兎耳山俊吾(富士通)

<タイトル>

通信ソフトウェア開発「短期リリース」への道

<サブタイトル>

～オフショア適用にも挑戦～

<発表者>

氏名（ふりがな）： 兎耳山 俊吾（とみやま しゅんご）

所属：富士通株式会社

<要旨>

今回発表するソフトウェア開発ターゲットのN F V（Network Functions Virtualization）市場は立ち上がり時期であり、N F V 標準もコンセプトレベルである。このため概念検証（PoC）で製品イメージを具体化し、お客様の声をフィードバックする開発プロセスが不可欠である。そこで今回「短期リリース」を目標とし「オフショア利活用」したソフト開発に取り組んだ。本発表ではソフト開発で実施した改善施策で適用した開発プロセスや作業定義を紹介する。最後に、改善施策後の課題や気づき、今後のソフト開発に向けて取組みを紹介する。

<キーワード>

ウォーターフォール開発、アジャイル開発、オフショア開発、短期リリース、通信ソフトウェア、N F V、概念検証、コミュニケーションツール、自動試験、工程定義、品質評価、フィードバック

<想定する聴衆>

ウォーターフォール開発者、アジャイル開発者、オフショア開発者、およびこれら開発に興味のある方

<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（次施策へのフィードバックが必要）

<発表内容>

1.背景

当社ソフト開発部門は通信キャリア向けソフト開発が主業務としており、ウォーターフォール型でのソフト開発を実施している。また、ソフト開発の課題であるコスト削減に対しては、オフショアの利活用を積極的に推進している。今回、NFVがターゲットのソフト開発プロジェクトが発足した。NFVとは、既存通信サービスを提供している専用ハード／専用ソフトで構成されたアプライアンス機器を、ITクラウドコンピューティング技術を活用し、大容量汎用サーバ・スイッチ・ストレージ上で仮想化した通信サービス機能として実装する技術である。NFVは市場が発展途上であるため、以下の特徴がある。一点目として、NFVの標準規約がコンセプト定義にとどまって具体的な要件まで落とし込まれていない状況がある。二点目として、NFV製品に対する顧客やベンダーにて、具体的な製品利用イメージが定まっていない。すなわちNFVのソフト開発においては、概念検証(PoC)等で製品イメージを具体化し、お客様の声を次回ソフト開発のインプットとしてフィードバックする「短期間」かつ「繰り返し型」での開発プロセスを適用する必要があった。

2.改善前の状態

今回のNFVソフト開発において、オフショア利活用した「短期リリース」による「繰り返し」開発プロセスを、従来のウォーターフォール型開発で適用した場合、開発製品のフィードバックについてはリリースまでの期間が長いこと、ソフト開発中にお客様からタイムリーに情報を得ることが難しいこと、また工程定義については、工程毎に前工程完了／移行判定や品質ふり返しを実施するため時間がかかることなど、短期リリースの開発には適していない、という課題があった。今回のNFVソフト開発でこの課題を解決するために、新しい開発プロセスを適用する事が必須となった。

3.改善前の状態をもたらした原因（因果関係）

上記の課題は、今回の開発要件とウォーターフォール型開発の特性とが適合しない事に原因がある。そこで「短期リリース」による「繰り返し」開発プロセスを実現するため、「アジャイル開発」に関して社内での開発事例調査を実施した。その結果（1）顧客と一体となった開発事例で適用、（2）開発規模が小さいプロジェクトで適用、（3）大部屋開発での適用、等の適用事例があった。（1）/（2）に関しては、いずれも今回の開発要件に適合しているが（3）を「オフショア」で実施する事は適用しづらい点があった。つまり、本開発でアジャイル開発、またはウォーターフォール開発のいずれか一方のみを選択する開発プロセスでは、いずれも適用しづらい開発プロセスが含まれている事が判明した。

4.計画した変更内容

上記検証結果より、「短期リリース」による「繰り返し」開発プロセスを「オフショア利活用」して実現するため、最適な開発プロセスの選択を試みた。具体的には、まず、本ソフト開発の開発プロセスを定義するために必要な作業項目を抽出した。次に、作業項目毎にウォーターフォール型開発とアジャイル開発での開発プロセス定義を記載し、今回のソフト開発プロセスを選択した。作業項目は、品質／検収部門の製品評価で提示が必要となる項目（工程定義、生産物定義、品質保証）今回のソフト開発特性からの項目（要件定義）およびオフショア利活用の項目（開発拠点形態）をあげた。作業項目毎の開発プロセス定義と開発プロセス定義の選択理由について、下記表に示す。

表 4-1 本ソフト開発での開発プロセス定義と開発プロセス定義選択理由

作業項目	選択開発プロセス定義	開発プロセス定義選択理由
工程定義	V字モデル定義 ← 工程定義レス	品質 / 検取部門に対し、従前の開発結果提示が必要だったため
生産物定義	工程毎定義 (↓KPT見直し後) 文書レス (ソース命)	短期リリースのため生産物作成量を最小限にするため
品質保証	工程毎品質保証 ← イテレーション内品質保証	品質保証はイテレーション内で2回実施、各工程毎の品質保証は実施しないため
要件定義	全要件を確定 ← リリース毎要件見直し	短期リリースのリリース結果を要件定義にフィードバックするため
開発拠点形態	海外拠点と協業 ← 国内大部屋開発	海外拠点 (オフショア) 開発優先のため

また、今回のソフト開発で、開発プロセスを実現するため適用したソフト開発ツールについて以下の図に示す。

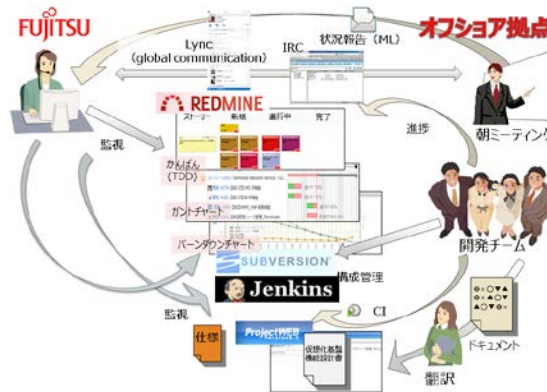


図 4-1 今回適用したソフト開発ツール

5. 変更の実現方法

短期リリースに向け選択した作業項目毎の開発プロセス定義については、KPT による見直しをリリース毎に実施し、見直し結果によっては、作業項目毎の開発プロセス定義の適用範囲を変更した。具体例として、生産物定義において開発当初は短期リリースを最優先と考え、生産物は最小限で作成する選択をした。ここで、初期イテレーションにて上流工程の仕様漏れ問題が発生し、仕様ドキュメントをすべて作成していなかったため問題分析ができない事例が発生した。本問題発生時の振り返りにより、極端な生産物レスは製品の品質保証ができないと判断し、以降のイテレーションでは、特に上流ドキュメントはもれなく作成する様、開発プロセス定義を変更した (表 4.1 緑矢印)。

また、バックログ見積作業に対しては、イテレーション毎にユーザーストーリーのタスクへの分割、およびタスクの作業時間に関する見積値と実績値のフィードバックを実施した。初期イテレーションでは、手さぐりでバックログを分割 / 見積を実施したため、バーンダウンチャートが右肩上がりとなる事態が発生した。KPT を使用して見積時のタスク分割の粗さと開発中のバックログ追加が、バーンダウンチャートが収束しない原因であると感じ、次イテレーションよりタスク分割粒度の細分化と、追加タスクの優先度により次イテレーションに回すフィードバックを実施した。これにより、2 回目以降のイテレーションから徐々に通常のバーンダウンチャートに収束した。

オフショア観点の改善例は、オフショア拠点とのコミュニケーション向上の手段として、富士通全社で導入されている Lync(global communication)を活用した点である。当初、コミュニケーション手段は ML と IRC、および Lync の InstantMessenger 機能と電話会議機能を使用していた。しかし、コミュニケーションミスによる仕様と異なる生産物が開発される事象が発生した。開発後半から Lync デスクトップ共有機能を使用開始した。拠点間の担当者は直接資料の共有が可能となり、打合せ時の資料への変更内容を即時反映できるようになった。このデスクトップ共有機能の活用によりオフショア拠点との精度の高いコミュニケーション改善が図ることができた。

6. 変更後の状態や改善効果

今回のソフト開発では当初の課題である短期リリースに対して、一ヶ月毎のリリースを厳守する事ができた。短期リリースが実現できた事で、リリース後の概念検証や総合試験の結果を次の開発へのインプットとしてフィードバック（計 39 件）し、これにより顧客からの要求に対して柔軟に対応できた。また、当初、ソフトウェアの短期リリースが開発効率と製品品質にどのような影響を及ぼすか不明だったが、今回のソフト開発での開発品質の値は、従前の開発プロジェクトとほぼ同等なレベルを確保した事も確認できた。

7. 改善活動の妥当性確認

今回のソフト開発は短期リリース実現をするため、各種開発プロセスに対する改善に取り組んだ。短期間リリースを厳守できた要因として、まず、ソフト開発の作業項目毎にウォーターフォール型開発とアジャイル開発での開発プロセス定義を記載し、今回のプロジェクトに対して最適な開発プロセスを選択した事が挙げられる。つぎに、開発プロセス運用中に出てきた課題を KPT 分析し、次リリースでの開発プロセスへ確実にフィードバックした事も成功要因である。

またオフショア利活用の点では、メールや電話会議システムといった従来からのコミュニケーションツールに加えて、Lync を利活用する事での、費用／時間のムダ削減とコミュニケーションでの情報量と精度を向上による改善効果が大きかった。さらに、各種アジャイルツール導入（Redmine でのかんばん運用、等）によっても費用／時間のムダ削減と進捗状況の精度向上の点で開発プロセスの改善に大きく寄与した。

今後の開発にあたっては、開発プロセス、契約プロセス、品質保証プロセスでの改善適用範囲の拡大について取り組んでいく。開発プロセスの改善は、自動試験適用範囲の拡大、および概念検証環境／顧客に対してタイムリーな製品提供とフィードバックを得るため、リリース期間の短縮（1ヶ月⇒2週間）に向けた施策を実行する。契約プロセスの改善は、本ソフト開発プロセスに適合した生産物定義に対する顧客納品物の理解度向上への取り組みを、品質保証プロセスの改善については、工程完了判定および開発完了判定の条件（品質指標／バグ密度、等）に対する定義内容の精査を実施予定である。

最後に、今回の通信ソフトウェア開発「短期リリース」への道を実践し、みなさんにお伝えしたい事は以下のとおりである。「自プロジェクトの業務を、深く分析し、改善を実践し、ふり返り、を繰り返して継続する事が、成功への道である。」

2A2 「海外開発拠点におけるソフトウェア開発プロセスの確立」 深井十三 (Panasonic Industrial Devices Singapore)

<タイトル>

海外開発拠点におけるソフトウェア開発プロセスの確立

<サブタイトル>

なし

<発表者>

氏名 (ふりがな) : 深井十三 (ふかいじゅうぞう)

所属 : Panasonic Industrial Devices Singapore, Singapore Technology Centre

<共同執筆者>

氏名 (ふりがな) : 佐古元彦 (さこもとひこ)

所属 : Panasonic Industrial Devices Singapore, Singapore Technology Centre

<要旨>

シンガポール開発技術センターでは 2002 年より組込みソフトウェアの開発を行っており、これまでは日本主導の下プロセス改善活動を行ってきた。しかし、シンガポールと日本の文化の違いにより意図した定着が進まなかった。そのためシンガポールの文化に合わせた改善が必要であると考え、日本でのプロセスを元に現地に即したプロセスのカスタマイズを行うと共に、コミュニケーションの課題を Redmine の導入により解決した事例を紹介する。

<キーワード>

海外、英語、文化の違い、コミュニケーション、多拠点開発、プロセス改善、テーラリング、Redmine、ワークフロー

<想定する聴衆>

海外での開発に携わる方、SEPG 初心者

<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

シンガポール開発技術センター(SGTC)は 2002 年に設立、日本、欧米、ASEAN の顧客に対して、日本の事業部との共同にて組み込みの車載ソフトウェアの開発を行ってきた。シンガポールの大きな特徴に、東京 23 区程度の非常に小さな国土にもかかわらず人口 500 万人を超える多民族国家であるという点が挙げられる。これは、シンガポールは過去から国として英語を公用語に外国人労働者を積極的に受け入れてきた結果であり、2013 年には人口全体に占める外国人の割合は華人、マレー人、インド人を始め 38.6%にまで達した。また世界有数の情報化先進国であり様々な国際ランキングで上位を占めると共に、国民へのインターネット普及率も非常に高い。

2.改善前の状態

シンガポールの公用語は英語で且つアジア随一の多民族国家であるため、様々な国（2015/6 時点 ASEAN を中心とした 9 カ国）のエンジニアが SGTC の開発プロジェクトに参加している。しかし全員が英語でコミュニケーションが取れるものの、皆母国語ではなく第二言語としての英語である上、それぞれが育ってきた国の背景や文化は大きく異なっていた。そのため異なる人種間での意思疎通が十分にできず、問題が起きることも少なくなかった。また、開発は日本側との共同開発や委託開発となるため、日本側の英語スキルの問題も重なり、日本との間でも同様にコミュニケーションロスが多発、特にプロジェクトマネージャの負荷が増大していた。すなわち「言語と文化の壁」によるコミュニケーションロスである。

更に、日本との遠隔での共同開発や委託開発から来る「距離の壁」による問題もあった。情報の過不足やタイムラグ、拠点間での情報の不整合により混乱が起き、また日本側からも現地の状況が見えなくなるという課題が発生していた。

開発プロセス面では日本の事業部との共同開発が中心となることを考慮し、過去から共通言語として日本と同一プロセスを導入し一体での開発を進めてきた。しかし、日本とのシンガポールとの間の文化の違いから現地で意図したような定着やその後の改善が進まず、形骸化するという事態に陥っていた。

一方、組み込みソフトに限らずソフト開発業界全体、特にオープンソースの開発を見てみると、言語や文化、距離を越えて世界中の人々が Web 上のみにて上手く開発を回していることが分かる。そこで、シンガポール国民の ITリテラシーの高さも踏まえて、オープンソース開発で使われている Web 上での仕組み、ツールを取り入れることでこれらの課題の解決を図った。

3.改善前の状態をもたらした原因（因果関係）

過去、トップダウンで日本のプロセスを導入することで早期に開発プロセスを立ち上げることができたが、日本との文化の違いもあり現地ではプロセスの意図を理解しないまま仕組みが実装されていた。またコミュニケーションのツールとしても未だエクセルやパワーポイント、Eメールに頼った運用で且つ、日本独特のやり方（すりあわせの文化）の下での開発を強いられることも多く現地でも困っていた。

そのため、今回はオープンソース開発でよく使われている Web 上での仕組みやツールを参考に、特に下記の点に絞って改善検討を行った。

- ・ 日本のプロセスを元にしたテーラリングの実施
- ・ 運用する仕組みの確立
 - 言語によるコミュニケーションの負荷を下げるツールの導入
 - 情報を集中管理するツールの導入

4.計画した変更内容

開発プロセスについては日本との共通言語となるため基本の考え方は変更せず、意図を変えない範囲でシンガポールでの組織、文化に合わせたプロセスを新たに制定し（テーラリングの実施）仕組みへの落とし込みを検討した。また、日本－シンガポール、現地内部でのコミュニケーションの円滑化と全ての情報の一括管理のために Redmine を新たに導入し、上記プロセスの Redmine 上への実装を計画した。

5.変更の実現方法

日本 Panasonic AIS 社開発標準である SPDR (System Product Development Rule) を元に SGTC に合わせたテーラリングの実施と組織プロセスを CMMI に基づき新たに制定。そして、これらの基本プロセスを Redmine 上に実装した。特に遠隔での開発における不必要なコミュニケーションから発生する混乱を防止するためにワークフローを必要最小限に単純化し、入力項目の必須化や非表示化を細かく設定した。また構成管理サーバ (SVN) との連携により全情報を Redmine 上から集中アクセスできるようにすることで情報のタイムラグや不整合が発生しないようにした。なお、海外での商用ソフトウェアの使用はライセンスの問題が起りやすいためライセンスソフトウェアは使用せずにシステムを構築。そして、これらのプロセスとシステムを継続的に改善、教育できるような体制にて運用を開始した。

6.変更後の状態や改善効果

Redmine の導入により全ての情報が一箇所に集約されることとなった。また、全てのタスクがワークフロー化され、誰が何をいつまでにするのが拠点を越えてリアルタイムで明確にでき、不必要な混乱を減らすことが出来た。例えばあるプロジェクトでは、これまでバグ管理について内部検出、日本事業部検出、カスタマ検出と3種類のエクセルのリストがあり E メールでやり取りしていたため管理が非常に大変であった。これを日本側の事業部も巻き込んで Redmine 上に一元管理することによって、全てのバグの対応状況がリアルタイムで全拠点から見えるようになり、プロジェクトマネージャーの負担も減らすことが出来た。また、基本のプロセスは日本側と共通としたことから、我々の Redmine への実装方法が日本側でも AIS 社標準の方法として採用された。

7.改善活動の妥当性確認

海外でのソフトウェア開発における課題である「言語、文化の壁」「距離の壁」をオープンソース開発で使われている仕組みを参考に解決できないかを考えた。海外では日本独特のすりあわせの文化が少ないため、開発のシステム化（ワークフロー化）は現地エンジニアに受け入れられやすく、その結果として開発効率が向上した。また、事例の中には逆に日本での開発フィードバックできるものもあるのではないかと感じている。今回は開発プロセス、コミュニケーションロスの課題に対して改善を行ったが、他の課題についてもオープンソース開発の仕組みを参考に解決できるのではないかと考えている。

2A3 「東芝製複合機組込みソフトウェア開発の海外拠点移管に関する取組み」 鈴木厚志(東芝テック)

グローバル開発力強化によるローコストオペレーション体制構築を目的として、東芝製複合機の組込みソフトウェア開発プロジェクトの大部分を海外拠点(インド)に移管する1年がかりのプロジェクト。現行の開発プロセスやインフラを再点検し、海外拠点を活用する協業プロセスの立案と体制構築を行い、複合機組込みソフトウェア開発の業務シフトを実現した。プロジェクトメンバにとっては、海外拠点活用に関しては未経験の領域であり、SPI 推進メンバの協力の下で、複数の課題解決やプロセスの定義を行い、海外移管を実現した。

<タイトル>

東芝製複合機組込みソフトウェア開発の海外拠点移管に関する取組み

<サブタイトル>

ありません

<発表者>

氏名(ふりがな)：鈴木 厚志(すずき あつし)

所属：東芝テック株式会社 プリンティングソリューション事業本部

<共同執筆者>

氏名(ふりがな)：加藤 優一(かとう ゆういち)

所属：東芝テック株式会社 プリンティングソリューション事業本部

氏名(ふりがな)：辻村 好弘(つじむら よしひろ)

所属：東芝テック株式会社 プリンティングソリューション事業本部

<要旨>

グローバルローコストオペレーション実現に向けて複合機組込みソフトウェア開発業務の大部分を海外拠点に移管するまでの手続きとして、海外拠点プロジェクトメンバへの OJT の実施、そしてトライアルプロジェクトを実施した。その中では様々な問題が検出されたが、PJ 関係メンバばかりでなく SPI メンバに協力をもらい課題を解決してきた。コミュニケーション基盤の異なる2か国において、意思疎通する事の難しさを体験し、これらを克服しながら進めてきた SPI 活動である。この成果として、海外拠点を活用する協業プロセスを作成し、新しい体制でプロジェクト完了する事ができた。

<キーワード>

業務プロセス、開発プロセス、開発プロジェクト

コミュニケーション、グローバル、海外拠点

ローコストオペレーション

組込みソフトウェア、複合機

<想定する聴衆>

SEPG、開発プロジェクトリーダー、組織長

<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（変更の結果が実現されている段階）

<発表内容>

1章 目的とゴール

海外拠点（Toshiba Software India Pvt. Ltd , 以下 TSIP）活用拡大による、コスト構造改革を目的とし、東芝製複合機 組込みソフトウェアの開発とプロジェクト運営を、TSIP を中心に実施できるようにする事が目的。国内拠点（東芝テック株式会社 三島事業所、以下 TTEC）と TSIP の垣根を取り払った、TSIP を中心とした東芝製複合機 組込みソフトウェア開発に関する協業プロセスの確立と業務シフトをゴールとした。

2章 進め方

2-1 準備

TSIP メンバに TTEC のプロジェクト推進の方法を教える準備ために、自分たちが行っている開発業務プロセス、体制、帳票類、ツール、インフラの点検を行い、日本語版でのドキュメント化を整備した。TSIP メンバに説明する為に英語翻訳を行った。

2-2 教育/OJT

教育の手段としては、最初のステップとして、電話会議、WEB を使った英語ドキュメントによる、各工程のプロセスや TTEC ソフトウェア社内認定のプロセスなどを説明。次のステップとして、プロジェクト業務を実際に体験してもらい、説明した内容と実業務のギャップを埋めてもらう為に、TSIP メンバ 2 名交替で、延べ 2 か月程度日本に滞在させ OJT を実施した。OJT の実施期間中についても、TSIP オフショアメンバと電話会議、WEB を使ったプロセス説明は継続して行った。TSIP に議事録を作成してもらい TTEC で議事録内容を確認するなど、説明した内容を理解できたか確認した。

2-3 プロセス定義

また、この TSIP オンサイトが日本に滞在している OJT 期間中に、TSIP を中心とした東芝製複合機 組込みソフトウェア開発プロセス（以下、組込みソフトウェア協業プロセス）を立案し、オンサイトメンバとオフショアメンバ、日本人メンバを交えて電話会議、WEB を使いレビューを重ね、協業プロセスを構築していった。2 国間の協業プロセス定義で必要なことは、コミュニケーションギャップを如何になくし、正確に同じ認識を持ってもらうことで、その点については、さまざまな工夫が必要となった。プロジェクトの社内専用用語については、コミュニケーションマップ（日英対応表）を作って、電話会議や説明資料では、このコミュニケーションマップを使う事で、正確に伝えるようにした。通常の日本国内、日本人同士での開発では、“伝えなくても、伝わる（High Context）文化であり、我々はこのことに慣れてしまっている。海外サイトと共に仕事をする際には、伝えないと、伝わらない（Low Context）事を認識し、注意しなければならない。

2-4 トライアルプロジェクト（プロセス検証）

定義した協業プロセスがうまく回るかを検証する為に、トライアルプロジェクト（実際には製品リリースをしない仮想プロジェクト）を実行した。トライアルプロジェクトの目的は、組込みソフトウェアの成果物を産出す事ではなく、協業プロセスを産出すことがゴールで、プロセスに欠陥が見つかった場合には、そこで一旦立ち止まりプロセスの改善を行ったうえでプロジェクトを再開するように進めた。TSIP 側はプロジェクトの推進を求め、TTEC はプロセスの確立を求めてしまった結果、一時はプロジェクト自体が迷走した時があったが、SPI メンバ

のバックアップのもと、プロジェクト目的の再確認と、お互いを理解する作業を根気よく行い、無事にトライアルプロジェクトが完了できた。

2-5 業務移管完了と実施

東芝製 複合機 組込みソフトウェアのバージョンアップ（ランニングチェンジ） 規模 200 人日
を TSIP 最初のバージョンアッププロジェクトとして実施し、予定通り完了する事ができた。
その後、同程度のプロジェクトを予定通り完了した。

3章 まとめ

OJT、トライアルプロジェクト、本番といった手順を踏んで、東芝製複合機 組込みソフトウェアの開発拠点の中心を国内中心から海外拠点（TSIP）にシフトし、開発費コストを大幅に改善する事ができた。また、プロジェクトメンバとしては今までに経験のない海外拠点活用という、我々にとっては難易度が高いテーマであったが、SPI 組織を活用する事によって、目標を達成できたものと考えている。

以下、海外拠点へのプロセス移管に関して留意する点を列举してみた。

- ・海外拠点への業務移管は、細かい部分まで文書化が必要。
- ・コミュニケーションギャップをなくすための、マッピングが必要。
- ・プロセス定義では、時差、インフラの考慮が大切。
- ・海外拠点の活用に関しては、お互いがお互いの文化や国民性を理解し、それを活かす事が大切。
- ・海外拠点のメンバの方が、プロジェクトマネジメントに優れたノウハウや仕組みを持っている場合があり、活用を検討する。

[今後の課題]

設計開発工数は全体の 80%を TSIP に移管できているが、プロジェクトマネジメント工数に関しては、20%未満程度の移管率である。そのため、プロジェクトマネジメント工数の海外拠点依存率を増加させることが、今後の課題である。この要因としては、組込みソフトウェアのバージョンアップは製品全体の一部の開発であり、国内拠点の多くのメンバと関わる必要がある。また、社内規定が日本語であり、さらに SW の認定プロセスや、認定組織が国内拠点をベースとして作成されたものであり、海外拠点移管を難しくしている。

1.背景

東芝テック株式会社中期計画に基づき、海外拠点活用の拡大によるコスト構造改革が目的。東芝製複合機の組込みソフトウェアの開発拠点を Toshiba Software India Pvt. Ltd シフトに伴う、開発プロセスの確立をプロジェクトメンバと SPI メンバで組織的に対応した。

2.改善前の状態

国内要員を中心とした開発では、コストが高く、競争力の改善が必要。また、国内要員は、次の世代の製品開発にリソースシフトを求められているが、改善前は達成できていない。

[期待効果]

現状の東芝製複合機 組込みソフトウェアバージョンアップ開発を、国内を中心とした開発から海外（TSIP）を中心とした開発にシフトする事により、人件費の差額のコストダウンが期待できる。
また、国内の開発リソースを次の世代の製品開発要員にリソースシフトする事ができる。

3.改善前の状態をもたらした原因（因果関係）

従来から東芝テック（株）の国内要員が中心の開発体制であった。プロジェクト運営に当たっては、ソフトウェア認定や開発プロセス自体が国内で実施する事を前提とした規定になっていた。

【事象・原因の洗い出し・選定・特定した方法】

これについては、現状の事実から明確であった。

4.計画した変更内容

TSIPを中心とした東芝製複合機 組込みソフトウェア開発プロセス（協業プロセス）を構築するにあたり、既存のプロセスを変更或いは、追加する方針で推進した。責任、体制、インフラ、各プロセス定義などで、欠陥抽出が見逃され、本番プロジェクトで想定していない問題が発覚するリスクが予想された。

このため、これをできるだけ排除するために、“プロセス欠陥検出と改善”、及び、“課題管理とコントロール方法”を事前に検討した。

【既存のプロセスを変更・追加する を選んだ理由】

既存のプロセスを廃止し新たに構築するよりも、既存のプロセスをベースにプロセスを再定義する、あるいは、足りない部分は追加する方が、開発拠点移管工数がかからないと考えた。

5.変更の実現方法

トライアルプロジェクトでプロセス欠陥をより効果的に摘出し、あるべきプロセスを作りプロジェクトにフィードバックする、監視体制を整えた。構成メンバは、各分野の主に管理層、SPI メンバ、プロセス有識者で構成され、トライアルプロジェクトを第三者視点で監視する。課題やプロセス欠陥を検出すると、課題登録簿に登録し、課題解決までコントロールする。

【工夫した点・苦労した点】

プロセスの欠陥は、主に海外拠点 TSIP（英語圏）と、東芝テック（日本語圏）のコミュニケーションギャップから発生する項目が多かった。これらを極力少なくするためにコミュニケーションマップ（用語対応表）を作成して、意思疎通が円滑にかつ間違いなく行えるように工夫した。検出された課題については、検出者やプロジェクト構成メンバだけでは解決できず、プロジェクトを取り巻く各方面の協力が必要であったが、拠点移管に関する認識が、国内関係メンバに浸透されていない場面も多くあり、協力をもらう為に苦労した。SPI メンバのバックアップをもらいながら、組織的に解決した。

6.変更後の状態や改善効果

課題やプロセス欠陥を修正する事により、行き違いや間違いの起きにくい協業プロセスを構築する事ができた。

【実現できたこと】

ソフトウェア設計の海外依存率を 20%上げることができた。また、プロジェクト運営に関わるプロジェクトスタッフの海外依存率を 0%から 20%に引き上げる事ができた。この結果、海外依存率の変化に応じた人件費コスト差額がコストダウンとして実現できた。シフトした国内要員は、新製品のコア開発にリソースをシフトする事ができた。

[測定方法・検証方法]

各プロジェクトの開発費工数を集計し、リソースの比率を検証した。

海外拠点活用比率			
	改善前	改善後	増減
設計	62%	82%	20%
プロジェクト	0%	19%	19%
設計評価	46%	91%	45%
合計	48%	76%	28%

7.改善活動の妥当性確認

[妥当性／費用対効果]

以下の点で妥当性があったと考えている。

実現方法の妥当性としては、1年未満の活動で、海外拠点を中心とするプロセスが定義でき、新しいプロセスを運用状態に持っていくことが出来たので妥当と考えている。

- (1) 定義したプロセス 92 工程 検出課題 110 件
- (2) 東芝製複合機 組込みソフトウェアのバージョンアップ効率

[残存課題]

200 人月程度のバージョンアップに対して実現するための課題を解決したが、数点の課題は残存課題として管理している。

その中の一つに、知財権調査の海外拠点移管である。特に日本語特許の知財権調査は、海外拠点メンバーには言語の壁があり、困難と考えている。英語特許の知財権調査に関しては、現在取組中であるが、トレーニングの実施や、責任の範囲の明確化が課題となっている。

また、SW 認定規定や、その他の認定規定は日本語ベースで、国内拠点をベースとしたプロセスになっている為、プロジェクトスタッフの業務について飛躍的に依存率を上げる為には、組織構造的な改革が必要。

2A4 「海外拠点におけるローカル講師育成による品質・プロセス教育の定着」 田村朱麗(東芝)

<タイトル> 海外拠点におけるローカル講師育成による品質・プロセス教育の定着

<サブタイトル>

<発表者>

氏名(ふりがな)： 田村 朱麗 (たむら しゅれい)

所属：株式会社東芝

<共同執筆者>

氏名(ふりがな)： 艸薙 匠 (くさなぎ たくみ)

所属：株式会社東芝

氏名(ふりがな)： 小島 昌一 (こじま しょういち)

所属：東芝ソフトウェア・コンサルティング株式会社

<要旨>

ある海外拠点で、自律的にピアレビュー教育を実施できるように、短期間でローカル講師を育成することにした。しかし、日本における従来の方法をそのまま適用するのは難しかった。そこで、効率的・効果的にローカル講師を育成し、教育を定着させる手法を考案した。本手法の特徴は、教えるポイントを明確にして教育テキストや講師向け教育等へ反映、通訳の質を上げるための工夫、文化背景の違いを考慮している点である。本手法により、短期間で6名の質の高いローカル講師を育成し、教育を定着させた。そして、ピアレビューの質が改善した。

<キーワード> 品質保証、品質管理、プロセス改善、人材育成、教育、トレーニング、海外拠点、グローバル展開

<想定する聴衆> 人材育成、プロセス改善、品質保証、海外拠点の活用推進などに携わる方

<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

筆者らは、プロセス改善の専門家として、国内外の当社グループ会社の品質保証強化やプロセス改善を推進する立場にある。

ある海外拠点（非英語圏）の品質は、顧客の受入テストの欠陥密度の遷移が安定しないなど、改善の余地があった。品質に関する因果関係分析の結果、その要因の1つとして、「ピアレビューの質の低さ」が挙げられた。具体的には(1)軽微な欠陥ばかりが指摘される、(2)ピアレビュー会議がうまくファシリテートされていない(レビューと著者が衝突する、コメントするメンバが偏っている、細かい点に関する議論に多くの時間を使ってしまう)等の課題があった。この課題を解決するための施策として、全技術者(約200名)に、ピアレビュー教育(2時間/回、約20名/回、全10回)を実施することにした。また、この海外拠点は、今後も技術者数が増える見込みであり、それに備えて自律的にピアレビュー教育を実施できるように、ローカル講師を育成し、教育を定着させることにした。なお、当社のピアレビュー教育の概要および構成は[1]にて紹介している通りである。また、ピアレビュー教育を管理者も含めた職場ぐるみ教育として実施した理由は、ピアレビューにより欠陥を除去し品質を向上させるには、「認知的不協和のために、自分の欠陥は自分では見つけられない。だから、ピアレビューは大事。」、「欠陥を作り込んだ担当者を責めない。」等のピアレビューの重要性、概念や実践方法を、組織文化の一部としていく必要があるためである。

2.改善前の状態

本節では、この海外拠点においてピアレビュー教育を定着させる上での課題について説明する。まず、筆者が日本から地理的に離れた海外拠点に出張して、毎回講師をするのは予算的に難しい。また、特に人間の“認知”にフォーカスしているピアレビュー教育は、ローカル言語で実施するのが効果的である。そこで、海外拠点で自律的に教育開催ができるよう数名のローカル講師を育成することにした。育成期間は、できるだけ短期間(1週間程度)であることが期待された。しかしながら、これまでのピアレビュー教育の講師育成方法をそのまま適用したのでは、短期間に海外拠点に数名のローカル講師を育成するのは難しかった。

3.改善前の状態をもたらした原因（因果関係）

これまでは、1度に育成する講師候補は、たいてい1名であり、OJT(On the Job Training)的に育成していた。教育開催頻度に依るが、育成期間は、1週間～3ヶ月程度だった。しかし、今回は数名の講師を、約1週間で育成したかった。つまり、従来の方法をそのまま適用するのは難しかった。また、教育テキストは英語、説明はローカル言語で実施することにしたため、その点も考慮する必要があった。

4.計画した変更内容

そこで、表1に示す手順でローカル講師を育成し、教育の定着させることにした。[手順0]～[手順4]の期間は、1週間程度を目安とした。先に述べた通り、教育の説明はローカル言語で実施することにしたが、筆者はローカル言語が分からない。通訳を介してコミュニケーションをとる場合、一文一意で重要な内容に絞ってシンプルに伝える必要がある。そこで、[準備]で、ピアレビュー教育で教えるべきポイントを明確にした。また、[手順1]～[手順4]において、通訳の質が大変重要になると考えた。そこで、[準備]と[手順0]で、通訳者の質を高めるための工夫をした。山本五十六の名言「やってみ

せ、言って聞かせて、させてみせ、ほめてやらねば、人は動かじ[2]]は、[手順 1]~[手順 2]、[手順 4]にマッピングできる。今回は、筆者とは文化背景の違うローカル講師候補を育成することから、[手順 1 やってみせる]で筆者が海外拠点の事情を十分に考慮することや、[手順 2 言って聞かせる]で、講師候補が筆者の期待通りに行間を読むことは、難しい可能性があった。また、[準備]で、対象組織の特性、ピアレビュー会議の様子、ピアレビューでの課題など把握するように努めた。リモートで情報収集するのは限界があると考えた。そこで、オンサイト([手順 0][手順 1][手順 2])で追加情報を集めた。それらを考慮した上で、[手順 3 もう1度、やってみせる]で、ローカル講師候補にお手本を見せることにした。[手順 5]では、今後、海外拠点で自律的にピアレビュー教育を実施できるように、新人教育へピアレビュー教育を取り込んだ。

表 2 ローカル講師の育成方法と教育の定着

準備	教育のポイントを明確化する
手順 0	通訳者と綿密に打合せを実施する
手順 1	やってみせる
手順 2	言って聞かせる
手順 3	もう1度、やってみせる
手順 4	させてみせて、褒める
手順 5	組織内に、定着させる

5. 変更の実現方法

全技術者向けに10回のピアレビュー教育を実施する中で、「4.計画した変更内容」を次のように実現し、ローカル講師を6名育成し、教育を定着させた。

準備：約2ヶ月

まず、表 2 に示す教育で教えるポイント「ピアレビューの5つの重要ポイント」と、表 3 に示す講師として必要なスキル「講師として必要な5つの項目」を明確化した。そして、これらを分かりやすく伝えることを意識して、教育テキストのテラーリング、講師向け教育のテキスト作成、教育テキストの日本語から英語への翻訳等を実施した。また、通訳者が教育内容を適切に理解することを目的に、教育内容の脚本(通訳向け)を作成した。

表 3 ピアレビューの5つの重要ポイント

1	レビューの目的/重要性
2	認知的不協和
3	ピアレビューの特徴
4	ピアレビューでの役割(ピアレビューリーダー、作成者、レビューア、記録者等)
5	ピアレビューでの定量的なデータの使い方

表 4 講師として必要な 5 つの項目

1	プレゼンテーション(受講者を向き、適切な声の大きさを、はっきり説明すること等)
2	受講生とのコミュニケーション(質問の適切な受け答え、アイコンタクト等)
3	資料に基づく説明
4	自分の経験に基づく具体的な事例の紹介
5	演習のファシリテーション

また、表 2 のピアレビューの重要ポイントと表 3 の講師として必要なスキルをもとに、技術力および講師力がピアレビュー講師として相応しいマネージャまたはプロジェクトリーダークラスを 6 名、ローカル講師候補として選抜した。

手順 0～手順 4： 8 日間

表 4 に示すスケジュールに従い、8 日間でピアレビュー教育を実施するとともに、ローカル講師を育成した。

表 5 ピアレビュー教育およびローカル講師育成スケジュール

	1 日目	2 日目	3 日目	4～8 日目
AM	—	[手順 1] 第 1 回 ピアレビュー教育	[手順 3] 第 2 回 ピアレビュー教育	[手順 4] ピアレビュー教育(6 回)
PM	[手順 0] 通訳者と綿密に打 合せ	[手順 2] 講師向け教育	[勉強時間]	[ローカル講師自主開催] ピアレビュー教育(2 回)

[手順 0 通訳者と綿密に打合せを実施する]では、3 名の通訳者に対して、[準備]で作成した教育内容の脚本(通訳向け)の内容を十分に説明し、通訳の品質向上に努めた。この中で、表 2 のピアレビューの 5 つの重要ポイントについても、丁寧に説明し、通訳者に内容を理解してもらった。

[手順 1 やって見せる]では、筆者が、第 1 回ピアレビュー教育を実施し、ローカル講師候補は、このピアレビュー教育を受講した。

[手順 2 言って聞かせる]では、筆者が、ローカル講師候補を対象に、講師向け教育を実施した。講師向け教育では、表 2 のピアレビューの重要ポイントを再確認するとともに、表 3 に示す講師として必要な次の 5 つの項目について説明した。また、ローカル講師候補と、海外拠点における現状のピアレビューの課題について、議論した。

[手順 3 もう 1 度、やってみせる]では、筆者が実施する第 2 回ピアレビュー教育を、ローカル講師候補が聴講、教え方を再確認した。先に述べた通り、[手順 3]は筆者とローカル講師の文化背景の違いを補完するために設けたものである。なお、[手順 2]で、複数の講師候補が重要だと認識している(改善の余地があると考えている)ピアレビューリーダーの役割等については、さらに手厚く説明した。

[手順 4 させてみせて、褒める]では、各ローカル講師候補が、各自、ピアレビュー教育を実施した。それに対し、筆者が、オブザーバーとして、各教育を観察し、講師として認定した。観察時には、表 2 に示すピアレビューの 5 つの重要ポイントを技術力として、表 3 に示す講師として必要な 5 つの項目を講師力として、観察シートを作成し、1 項目 10 点満点、全体で 10 項目 100 点満点として評価した。評価結果は、教育実施後に良くできている点・改善すべき点などのコメントと

もに講師候補にフィードバックした。6名全員が高得点で合格した。なお、残り2回のピアレビュー教育は、ローカル講師が自主開催した。

手順5：5日間

[手順5 組織内に、定着させる]では、海外拠点の新人向けピアレビュー教育を、本教育をベースとした内容に改訂した。ローカル講師が、速やかに改訂作業を実施し、既に改訂した教育テキストを利用して新人向けピアレビュー教育が実施されている。

6. 変更後の状態や改善効果

本節では、育成期間とローカル講師の質、およびピアレビューの改善状況の観点で、「5. 変更の実現方法」の実践結果を評価した内容について説明する。

6.1 育成期間とローカル講師の質

<育成期間>

「4. 計画した変更内容」で述べた手法を適用し、全技術者向けに10回のピアレビュー教育を実施することを通して、全体で8日間、手順1～手順4に限定すると6日間で、6名のローカル講師を育成した。これは、「約1週間で、数名のローカル講師を育成したい」という当初の期待に合うものである。従来手法で講師育成をしたとすると、少なくとも2倍程度の期間/工数を要したと予想される。つまり、育成期間を半減できた。

<ローカル講師の質>

10回のピアレビュー教育の事後アンケートの結果における総合評価(4段階)の最高評価は60.4%であった。これは、筆者が国内で実施しているピアレビュー教育の事後アンケート結果とほぼ同等である。筆者の所属部門が同じ海外拠点に対して提供している複数の教育の中でも、比較的評価が高いグループに属していた。また、筆者が実施した2回と、ローカル講師が実施した8回の受講者事後アンケート結果は、仮説検定で有意差がなかった(データが正規分布に従っていなかったため、ノンパラメトリック検定の1つであるMann-Whitney U検定を利用した)。すなわち、筆者とローカル講師とで、ピアレビュー教育の質に差はなかった。このように、「4. 計画した変更内容」で説明したローカル講師を育成する手法により、短期間で質の高い講師を育成することができた。そして、定着のために、新人教育に取り込み、既に新人向けにピアレビュー教育を実施している(数回開催(2015年9月現在))。

6.2 ピアレビューの改善状況

「1. 背景」で述べた通り、ピアレビュー教育は、「ピアレビューの質の低さ (1)軽微な欠陥ばかりが指摘される、(2)ピアレビュー会議がうまくファシリテートされていない」を改善するための施策として実施した。本節では、ピアレビュー会議への参加による観察と、教育実施前後のピアレビューデータ比較分析により、この2つの観点の改善状況を説明する。

<ピアレビュー会議への参加による観察>

筆者は、この海外拠点の全技術者へのピアレビュー教育実施後、複数のプロジェクトのピアレビュー会議に参加した。その観察事項は図1であり、状況が改善していることを確認した。

- 観察事項(Good Points)
 - 軽微でない欠陥が指摘されていた。
 - プロジェクトリーダーが、ピアレビューリーダーとしてピアレビュー会議をファシリテーションしていた。
 - レビュー全員が、コメントをしていた。
 - ピアレビュー会議が、時間通りに開始されていた。
- 「軽微な欠陥ばかりが指摘される」状況が改善
- 「ピアレビュー会議のファシリテーション」が改善

図 1 ピアレビュー会議の観察事項

また、教育実施後、ピアレビュー会議他、品質問題について議論する会議において、参加者同士が「責めてはいけないよね。」と互いにいさめたり、「認知的不協和だから、自分では気づかなかった。」と発言したりと、ピアレビュー教育で教えた内容を意識して行動するようになった。

<教育実施前後のピアレビューデータの比較>

全技術者にピアレビュー教育を実施した前後でのデータを比較分析することで、ピアレビューの効果を定量的に確認した。具体的には、顧客が類似している教育実施前後のプロジェクトデータ(複数)を抽出し、レビュー効率¹とレビュー効率(軽微な欠陥を除く)²の平均/中央値を比較した。レビュー効率とレビュー効率(軽微な欠陥を除く)のいずれの値も、教育実施前よりも実施後の方が、約 50%高くなっていた。レビュー効率の向上については、「ピアレビュー会議のファシリテーション」が改善し、より効率的にレビュー指摘が実施された結果だと解釈できる。レビュー効率(軽微な欠陥を除く)の向上については、「軽微な欠陥ばかりが指摘されている」状況が改善されたことを示していると言える。

7.改善活動の妥当性確認

ある海外拠点において、自律的にピアレビューを実施できるように、ローカル講師を育成し、教育を定着させることにした。しかし、従来の方法をそのまま適用したのでは、短期間で海外拠点のローカル講師を育成するのは難しかった。そこで、効率的・効果的にローカル講師を育成し、教育を定着させる手法を考案した。本手法の特徴は、(1)教えるポイントを明確にして教育テキストや講師向け教育等へ反映、(2)通訳の質を上げるための工夫、(3)育成手順において筆者と講師候補の文化背景の違いを考慮していること、が挙げられる。(1)は通訳を介してコミュニケーションをとる場合、重要な内容に絞ってシンプルに伝える必要性から取り入れたものであるが、改めて考えると、国内で教育実施・講師育成をする際にも有効だと考える。(2)は通訳の質、(3)は文化背景の違いにフォーカスしており、海外拠点で教育実施・ローカル講師を育成する際の特徴だと言える。このように、本手法により育成手順の明確化/育成内容の形式知化し、短期間で、6名の質の高いローカル講師を育成した。従来手法で講師育成をしたとすると、少なくとも2倍程度の期間/工数を要したと予想される。つまり、育成期間を半減できた。ローカル講師は、期待通り、ローカル言語の適切な単語/表現を利用して教育を実施している。また、ピアレビュー会議参加による直接観察および教育実施前後のピアレビューデータを比較分析することにより、「ピアレビューの質」が改善していることを確認した。

ローカル講師を育成した副次効果としては、「ローカル講師は、海外拠点の身近な成功事例、失敗事例を説明することができるので、受講者のピアレビューの重要性に対する認識が高まる。」等が挙げられる。

今後は、受入テスト欠陥密度等の定量的データにより、「ピアレビューの質」の改善が、この海外拠点が開発するソフトウェアの品質向上に役立っていることを確認する。また、考案した手法を洗練し、ピアレビューだけでなく、他の品質・プロセス教育の海外展開にも適用していく。併せて、この手法の考え方を、日本における品質・プロセス教育の講師育成方法にも

¹ レビュー効率 = (レビューで指摘した欠陥数) / (レビュー工数)

² レビュー効率(軽微な欠陥を除く) = (レビューで指摘した欠陥数(軽微な欠陥を除く)) / (レビュー工数)

取り入れることで、日本における講師育成方法の効率化を図る。例えば、「4.計画した変更内容」[手順 4]の観察シートは、日本国内で講師を育成する際にも有効だと思われる。

<参考文献>

[1] はじめよう、レビュー活動!: <http://thinkit.co.jp/book/2009/03/02/550>, 小笠原秀人, Think IT, 2009

[2] 山本五十六の名言集「人を動かす」: <http://www.ym56.net/meigen.html>

2B1 「埋まっている会社の資産（人材）を最大限に活用する」長橋敦(シナジーテック)

世の中が景気回復の兆しを見せてきた昨今、開発に携わるエンジニアの確保が難しくなってきた。そこで、色々な理由で辞めた人や、休職中の人材を有効に活用することで、プロジェクトの人員不足を補う為のモデルである。今回は、産休中や育児休暇中のエンジニアを対象に実施してみました。

<タイトル>

埋まっている会社の資産（人材）を最大限に活用する

<サブタイトル>

なし

<発表者>

氏名（ふりがな）： 長橋 敦（ながはし あつし）
所属：株式会社シナジーテック ※ソフトハウスです

<共同執筆者>

なし

<要旨>

世の中が景気回復の兆しを見せてきた昨今、開発に携わるエンジニアの確保が難しくなってきた。そこで、色々な理由で辞めた人や、休職中の人材を有効に活用することで、プロジェクトの人員不足を補う為のモデルである。

※プロセス改善というよりは、ワーキングモデルに近いかもしれません。

<キーワード>

人材不足、人材登用、在宅、VPN、仮想環境、ワーキングモデル

<想定する聴衆>

プロジェクトマネージャー、プロジェクトリーダー

<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（実践した結果の事例報告）

<発表内容>

1.背景

ここ数年日本の経済が活性化してきているなか、ソフトウェア業界も例外なく忙しくなっている。しかしながら、リーマンショック時に人員整理（早期退職など）を行った影響か、はたまた、ソフトウェア業界の作業が過酷なせいか、ソフトウェア開発を行う「開発者」の確保が難しくなっている（世の中全体的に減ってきている？）。その人手不足をなんとか解消できないかと思い、本プロセスを試してみました。

本プロセスは、今回突発的に思いついたものではなく、かねてから、「世の中に埋まっている資産（シニアエンジニア、寿退社したエンジニア、産休中/育児休暇中のエンジニア）などを活用できないか」といったぼんやりとした観点を具体化したものである。

2.改善前の状態

プロジェクトの体制を決めるにあたり、「人員確保の見通しが付かない」状況にありました。今までは、なんとか体制を決める為に、

- スキルのハードルを下げて人員確保を行う
- プロジェクトのメンバーの作業ボリュームを増やす（1人複数役割）

といった、不安の残る対応を行ってきました。

3.改善前の状態をもたらした原因（因果関係）

「1」でも書きましたが、

- 数年前の「人員整理（早期退職など）」により、社内のエンジニアが減っている
 - アウトソースを期待するも、世の中的に景気が良くなり、思ったような人材が確保できない
- といった部分が原因の1つとしてあると思います。

4.計画した変更内容

なかなか人が集まらない中、イメージとして湧いてくるのは、過去に現場と一緒に作業していた人のことばかりでした。「あ〜、x xさんがいてくれたらなあ」とか「y yさん、仕事手伝ってくれないかなあ」など、「ないものねだり」をしている自分がいたのです。しかしながら、色々な事情で現場を離れている人が手伝ってもらえるはずがないのですが、このインフラの発達した世の中で、何かできないか と考えました（前述にも記述しましたが、この辺は昔からぼんやりと考えてはいました）。

そこで、思いついたのが、「早期退職されたシニアエンジニア」、「寿退社した女性エンジニア」、「現在産休中/育児休暇中の女性エンジニア」だったのです。これらの人に、現場まで来てもらわなくても「在宅」で作業を手伝ってもらうことはできないだろうか？と考えました。

在宅といった場合、真っ先に思いつくのが「情報漏えい」や「セキュリティ」などでした。（企業によっては、「社内コンプライアンス」も関わってくるかと思いますが）

5.変更の実現方法

在宅での「情報漏えい」「セキュリティ」をクリアする方法について幾つか検討してみました。

■情報を持ち出さない

→ 「VPN」を使用し「在宅先から社内情報を参照してもらう」ということでクリアできるのではないかと。

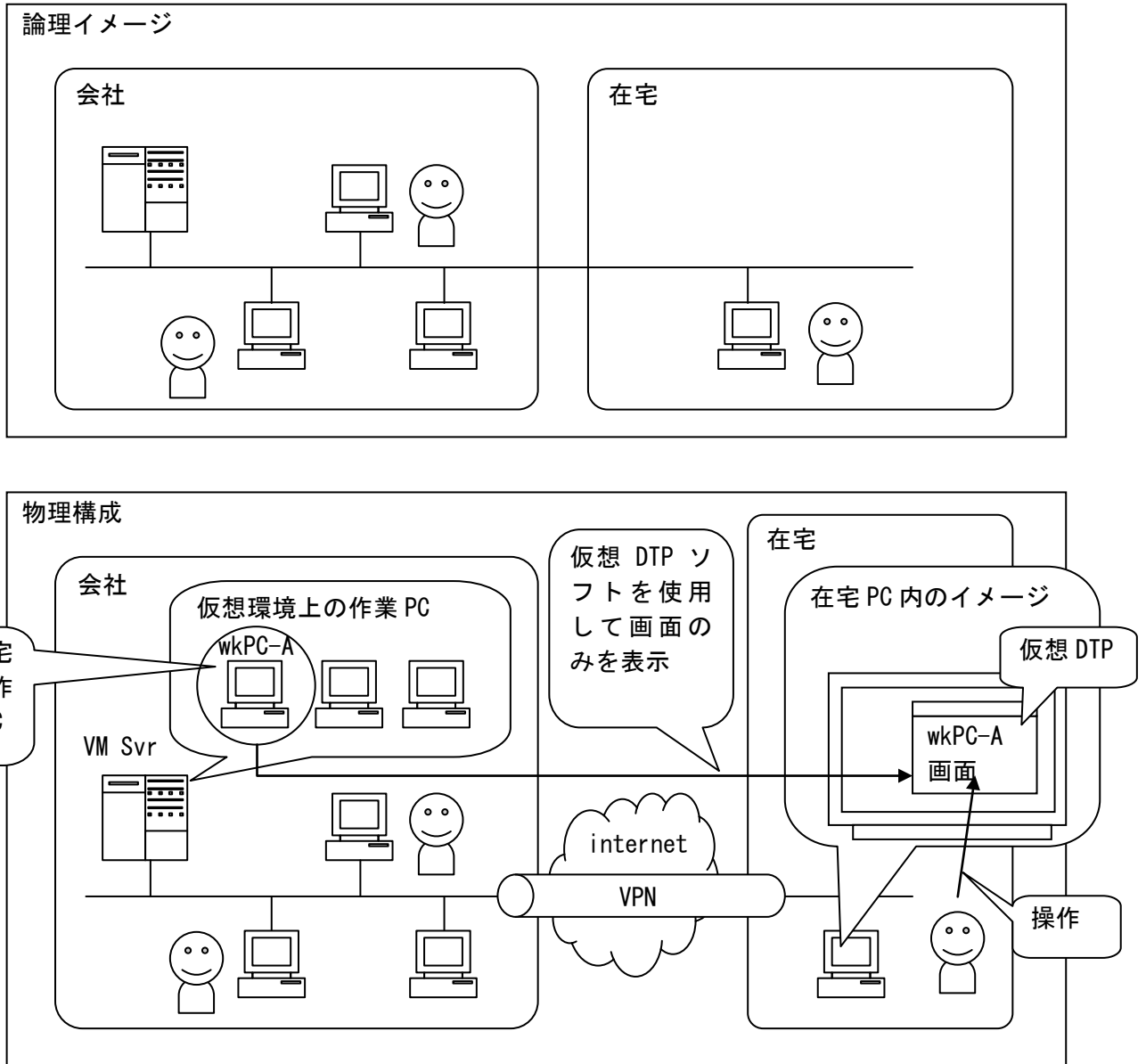
■社内の作業端末を貸し出せない

→ 「社内にある仮想環境上の PC」を使用することでクリアできるのではないか。

■ 情報開示に対する不安

→ 「会社関係者」に限定してしまえばクリアできるのではないか。

という結論に達しました。上記観点をイメージした図を以下に記述します。



上記により、「情報漏えい」「セキュリティ」に関する懸念点は大幅に減りました。

また、このような作業形態の場合、どのような作業を依頼するか検討したところ、「テストフェーズ」が一番インパクトありそうだったので、今回の作業は

- テスト仕様書の作成
 - 単体テスト（テスト仕様書をもとに）※エビデンスの確保も含むの部分
- を依頼しました。

テストフェーズに絞った理由は、「決定事項に対して、作業を行う」という観点からです。設計や実装の場合、なんだかんだいっても、不明点が発生し、その都度打ち合わせが必要となる。しかしながら、在宅の状態ではそれは難しい。しかしながら、「テスト仕様書作成」や「単体テスト」といった作業であれば、INPUT をもとに作業を黙々

と進めることが可能であると判断したからです。

6. 変更後の状態や改善効果

今回作業してもらったのは、「弊社社員の奥さん（仮に A さん）」に作業を依頼しました。A さんの現在の状況は以下の通りでした。

- エンジニア経験 4 年
- 産休中（子供は 6 ヶ月目）
- 少しでもいいから働きたい（収入を増やしたい）

本プロセスの内容を A さんに話、相談したところ、当方の思惑と、A さんの思惑が合致した為、本プロセスを実施してみました。実施した感想を以下に記述します。

■ 人材の確保

経験値のある人材を確保することが出来た。

■ 仕様説明がスムーズ

既に経験値があるので、仕様についても全てを説明するのではなく、概要のみを伝え、あとは資料を読んでもらいました。1 ~ 10 の説明が必要なところ、1、3、5、10 くらいで済んだ感じです。（当然、あとから質問はありました）

■ 開発プロセスの理解

経験者なので、開発プロセスの理解も早く、そのプロセスに則って作業してもらうことが出来た。（これらも、多くの説明を必要としなかった）

今回は、構成管理 Tool を用いた「トレーサビリティテストプロセス」を実施してみた。（考案後、初挑戦！）

■ 各種 Tool を駆使

経験者なので、作業効率が良いと思える Tool を自分で駆使して（マクロやバッチファイルを作るなどして）、作業効率や品質などの向上に努めてくれた。

■ 低予算

A さんは、子供が小さく、家から出て仕事ができなかったのが、時間単価が安くても「願ったり、叶ったり」だったようです。（時間単価は世の中のアルバイト金額 + αレベル）

■ その他

実際に A さんが、作業現場に足を運んだのは、説明をする為に来てもらった「最初の 1 回だけ」でした。その後は、メールやチャット、PC ソフトの音声通話で済んでしまいました。

7.改善活動の妥当性確認

プロジェクトが終了し、本プロセスを振り返った時の印象としては、「とても助かった」という印象が第一印象でした。これは、作業に対しても、予算に対しても、品質に対しても であります。しかしながら、懸念点（対策点）もいくつか出てきました。その内容は、

■作業ボリュームの提示

Aさんは経験者（エンジニア）だったので、「このくらいの作業はこなせるだろう」と当初考えていたが、小さい子供がいる為、突発的な事象が発生しました（作業時間が減る）。そういった、作業者の状況を考慮したボリュームを検討する必要がある。

■作業時間の把握

在宅（見えない場所での作業）なので、時間単価での作業となると「自己申告」となります（今回も自己申告でした）。作業時間をシビアに意識したい場合は、作業ボリュームに対して金額設定する方が良いかもしれません。

■不具合管理（準備不足）

単体テストをお願いしたにもかかわらず、不具合発生時のプロセスを提示しなかったことで、本来のプロセスでない作業（資料）が発生してしまい、フィードバックに手間取ってしまった。

以下、当方の感想

今回のプロセスを行ってみて、世の中に埋まっているリソース（人材やスキル）を上手に使える事がわかった。このような仕組みを企業が組み込んでいくことで、世の中における少子高齢化対策を少し「援護」する事が出来るのかもしれない。

2B2 「7社統合による品質マネジメントシステム（QMS）の再構築」 杉野晴江(NECソリューションイノベータ)

<タイトル> 7社統合による品質マネジメントシステム（QMS）の再構築

<サブタイトル> ～標準プロセスを示す新たな QMS の姿～

<発表者>

氏名（ふりがな）： 石垣 篤史（いしがき あつし）
所属： NECソリューションイノベータ株式会社 技術統括本部

<共同執筆者>

氏名（ふりがな）： 杉野 晴江（すぎの はるえ）
所属： NECソリューションイノベータ株式会社 技術統括本部
氏名（ふりがな）： 舟山 正憲（ふなやま まさのり）
所属： NECソリューションイノベータ株式会社 技術統括本部

<要旨>

7社統合により、「誰がどこにいても同じようなやり方で仕事ができる」をコンセプトに掲げ、品質マネジメントシステム（QMS）を統合した。統合 QMS はプロセス、アクティビティ、タスクという構造の定義と、プロセスフローダイアグラムによるプロセス表示方式を新たに採用し、Web ページとして作成することにより参照容易性を高めるよう工夫した。本年 4 月にリリースし、現在浸透を図っている段階だが、統合 QMS 構築によりプロジェクトの実績データの収集基盤ができ、データに基づく組織的プロセス改善が可能な状況ができあがった。

<キーワード>

品質マネジメントシステム（QMS）、統合、プロセス、アクティビティ、タスク、プロセスフローダイアグラム、Web、リンク、ワーキング・グループ（WG）

<想定する聴衆>

QMS の統合・QMS の再構築に興味のある組織の方（SEPG の方）

<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

2014年4月にNECソフト、NECシステムテクノロジー、北海道日本電気ソフトウェア、NECソフトウェア東北、北陸日本電気ソフトウェア、中部日本電気ソフトウェア、九州日本電気ソフトウェアの7社が統合され、当社（NECソリューションイノベータ）が誕生した。NECソリューションイノベータは組織体制の見直しや業務システムの統合と合わせて、1年以内に品質マネジメントシステム（QMS）を統合することを決定した。

2.改善前の状態

統合前の7社にはそれぞれ自社のQMSがあり、各社がISO9001の認証を取得していた。しかし7個のQMSは、ISO9001要求事項のプロセス実装方法やプロセス記述粒度、ISO9001要求事項以外のプロセス定義範囲などに相違点が多く、7社統合の直後は「一国七制度」とも言うべき状態であった。7個のQMSを統合したQMSを構築することにより、「誰がどこにいても同じようなやり方で仕事ができる」状態を実現することが期待された。

3.改善前の状態をもたらした原因（因果関係）

SI（システムインテグレーション）、ソフトウェア開発、サービスという業務プロジェクトの各工程、およびプロジェクトの管理プロセスは各々のQMSで定義されており、統合前の各社のQMSを適用して業務を遂行することは、統合会社全体として業務効率、リソース流動性、品質向上を阻害する要因であることは明らかであった。

4.計画した変更内容

統合QMSは「誰がどこにいても同じようなやり方で仕事ができる」をコンセプトに掲げ、1)7個の既存QMSを前提とせず、新たな会社組織体制を踏まえQMSを構築する、2)ISO9001要求事項だけではなくプロジェクトがやらなければならないことも網羅する、3)QMSの刷新をアピールするため、新たな表記・表現手段を採用すること、を統合QMS構築の大方針とした。これは、統合前の会社間の社員の心理的な壁を取り払い、統合会社として一致団結していく基盤を構築するとの考えを具体化したものである。

5.変更の実現方法

統合QMSの構築作業には、「基本設計」、「機能設計」、「執筆」、「試行」という、通常のシステム構築と同様のフェーズを設定した。基本設計ではQMS全体の枠組みの検討、およびプロセス、アクティビティ、タスクという構造の定義とプロセスの分割を行った。機能設計では各プロセスの活動内容をアクティビティとして定義するとともに、各アクティビティのインプットとアウトプットを定義した。この段階で、プロセスを図示する手段として多数の既存QMSで使用していたフローチャートではなくプロセスフローダイアグラムを新たに採用し、プロセスを構成するアクティビティとインプット、アウトプットの関係を図示した。執筆段階で各プロセスのアクティビティに関してタスクの内容を定義して、プロセスフローダイアグラム、アクティビティ、インプット、アウトプット、タスクをプロセス記述としてまとめた。また、統合QMSはWebページとして作成することにより、ファイルをダウンロードしなくても必要なプロセスのプロセス記述を直接参照可能とすると同時に、プロセス記述間をハイパーリンクで接続して参照容易性を高めた。試行段階では事業部門にプロセスの定義、記述、運用などを検証してもらい、検証結果を統合QMSにフィードバックして最終版を完成させた。

統合QMSの構築作業と並行して、各事業部門の代表者と構築チームでワーキング・グループ（WG）を立ち上げ、定期的に構築作業の進捗状況の確認や課題の共有、さらに中間成果物のレビューを実施した。

6. 変更後の状態や改善効果

当初の予定通り 2015 年 4 月に統合 QMS をリリースした。4 月、5 月は従来の QMS からの移行期間とし、新規発足するプロジェクトには統合 QMS の適用を、前年度からの継続プロジェクトにはこの期間中に統合 QMS への切り替えを指示した。5 月末時点で約 3,200 件のプロジェクトが登録されており、6 月以降は全プロジェクトが統合 QMS の適用で一本化されている。6 月、7 月を徹底期間と位置付けて、現在は統合 QMS の適用状況を確認しながら、事業部門およびプロジェクトへの浸透を図っている段階である。

具体的な効果の分析はこれからだが、統合 QMS の構築により、年間 10,000 件近くのプロジェクトの実績データを収集可能な基盤が整備され、データ分析による組織的プロセス改善が可能な状況ができあがった。

7. 改善活動の妥当性確認

1 年以内に統合会社全体に適用可能な統合 QMS を構築するという目標は達成できた。WG 活動を通して事業部門と協調しながら進めることによって、従来の QMS からの切り替えもスムーズに実現できた。本格的に浸透していくのはこれからであり、事業部門での運用に基づいて実績データを収集、分析して継続的なプロセス改善のサイクルを軌道に乗せることが課題である。

2B3 「システマティック類推見積 - 過去事例との新しい出会い」 高橋一郎(日立製作所)

<タイトル>

システマティック類推見積 - 過去事例との新しい出会い

<サブタイトル>

全社データとの相性診断！ あなたの見積を事例3000とマッチング！

<発表者>

氏名（ふりがな）： 高橋 一郎

所属：日立製作所 情報・通信システム社 生産技術本部 計画部

<共同執筆者>

氏名（ふりがな）： 清水 健一

所属：日立製作所 情報・通信システム社 生産技術本部 計画部

<要旨>

プロジェクトのポストモータムデータは、単に統計情報にしてしまうのではなく、見積プロジェクトとできるだけ近い条件のプロジェクトを探せるデータベースとして実例を用意すると、類推見積に役立つ。また、統計情報として百分順位を利用すると、全社の過去事例から見た自分の見積コストや見積工期の位置づけが理解でき、プロジェクトのリスクを鑑みた妥当性を判断するのに役立つ。

<キーワード>

類推見積、プロセス資産、測定リポジトリ、リスク管理、ポストモータム

<想定する聴衆>

ソフトウェア開発見積関係者

<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

我々は全社の生産性改善を推進する立場であり、頻発するトラブルプロジェクトに対し、対抗策を考える必要があった。

2.改善前の状態

大きなトラブルプロジェクト事例を調査し、プロジェクトのどの時点から問題が潜在していたのかを分析した。その結果、多くのプロジェクトは見積時に事例調査の不足などから失敗をしており、それを取り戻せずにトラブルとなっていたことがわかった。

3.改善前の状態をもたらした原因（因果関係）

見積失敗の原因のひとつとして注目したのが、生産性見積のミスであった。生産性はプロジェクトの条件に依存し大きく変動するのに、そのことをしっかりと理解していない見積者が少なくなかった。類似案件の事例をきちんと調べずに、生産性を平均値に基づいて設定しているケースや、まったく違う条件の事例に基づいて設定しているケースがあった。

4.計画した変更内容

見積者が類似案件の事例を十分に調べていないのには事情があった。プロジェクトのポストモテムデータは、すべてプロセス資産として Web でアクセスできるデータベース上に蓄積されていたが、その中から適切な事例を検索するのが困難だった。そこで、見積専用のリポジトリを新設し、見積者が適切な類推プロジェクトを検索したり、全社の過去データと百分順位で比較したりできるツールとして提供することにした。

5.変更の実現方法

全社の過去データと百分順位で比較するというコンセプトは、規模を基準にしたコストの百分順位（コストポジション）とコストを基準にした工期の百分順位（工期ポジション）のふたつで実現した。このため、このリポジトリはさまざまな条件の百分順位情報を保管する百分順位専用の測定リポジトリという側面がある。コストポジションは工程範囲、言語、契約形態、システム形態等、さまざまな条件を設定し、それぞれの条件での百分順位を出せるようにした。ただ、それだけだと良し悪しが分からないため、プロジェクトのリスクの大きさという切り口で、適切な値の範囲を示し、コスト見積の妥当性を検証できるようにした。工期ポジションもリスクの大きさにより適切な範囲が異なることが分かったため、適切な範囲に入っているかどうかを確認できるようにした。また同時に、入力された条件で類似プロジェクトを検索し、ポストモテムデータを参照したり、見積と比較したりできるようにした。

ただしこの分析は、結果を出すのはごく短時間でできるが、その内容を読み解くのはコツがいる。そのため、初めて見積レビュー用に分析を実施した人たちに対し、当方からその分析に対するコメントをメールで送り、注意点を助言するようになった。この活動は「分析結果分析」と呼んでおり、現在も継続している。

6.変更後の状態や改善効果

見積の事例調査不足を感じている上位管理者は多く、このツールは導入後半年で見積時の必須ツールに指定された。また実際、このツールを使用する時に、複数の事例を参照している見積者が多くあり、部署の枠を超えた事例の共有が進んできている。

7.改善活動の妥当性確認

生産性の分布について、見積者が現実の姿を認識することができ、類似プロジェクトを見つける手段を得たことは、見積改善に向けた大きな成果といえる。ROI を説明するのは難しい施策だが、例えばもし質問が「1億円の赤字を出すであろう見積を見直させることができたか」ということであれば、そこその確度で Yes と言うことができる。と言うのは、多くの見積者が使用後に見積を見直しており、見積を熟考するためのツールとして活用されていることがわかるためである。

2B4 「昔から使ってるので今更変えたくないんだけど」 斎藤学(ウイングアーク 1st)

会社組織の統廃合に伴いシステム、プロセスを統廃合する必要に迫られる場面は少なく無いと思われます。我々が2年間かけて実践し、成功した事例の発表です。

<タイトル>

昔から使ってるので今更変えたくないんだけど

<サブタイトル>

組織の統合によるシステム、プロセスの整理といつの間にか前より便利になっているを目指して

<発表者>

氏名（ふりがな）：斎藤学（さいとうまなぶ）

所属：ウイングアーク 1st 株式会社開発本部 SVF 開発統括部プロジェクトマネジメント支援グループ

<共同執筆者>

氏名（ふりがな）：

所属：

<要旨>

製品のリリースと一概に言っても、企画、計画、開発、検証、メディアの作成、ドキュメントの作成、販売、ユーザー対応、保守等多くの部署が関わります。ドキュメントそのものや、ドキュメントの管理やプロセスフローについてもそれぞれの部署で異なるやり方で行われている場合が多々あります。そのような異なる組織間でのやり方の違いをどのように統一してきたかの事例発表です。

<キーワード>

組織統合、システム統合、JIRA、Confluence

<想定する聴衆>

高成熟度組織の方、PM、PMO 等

<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

弊社ウイングアーク 1st は SVF という帳票製品を開発、販売しています。

開発部門とそれ以外とが考え方の違いにより、成果物の内容、作成までのプロセス、管理方法も異なっていました。開発部門でも複数のプロジェクトがありますので、プロジェクトごとにそれぞれ異なっていました。



fig.1：部門が違うと対応も違う

2.改善前の状態

開発部門では、仕様書やテスト報告書等のドキュメントがテキストファイルや Excel ファイルで作成、管理されていました。それらのファイルでの管理には多くの問題がありました。

例えば、ドキュメントの内容そのものの問題、ファイル管理の問題（最新版はどれ？誰が持っている？）等。



Fig.2:どれを見ればいいのか？

また、成果物を運用するシステムも部門ごとにそれぞれ別々に運用されていました。部門間での情報のやりとりも基本的にメール、Excel ファイルのやりとりとなっており、誰が最新の情報を持っているのか、誰の作業待ちなのかがわかりにくい状態でした。結果として部門間の調整会議や関係するステークホルダーが多くなっていました。

それらを改善することにより、ファイル、情報管理コストの軽減、問題解決までの時間の軽減、部門間調整会議を少なくする事や、そもそも会議に参加する必要が無くなることを期待しました。

課題はあるにしても、上手く回っているように見えるのならば敢えて変えたくはない、変える必要性を感じないという風潮もありました。

3.改善前の状態をもたらした原因（因果関係）

もともと複数の会社が合併した事により、組織、文化が異なる事が大きな要因です。部門内の作業ステータスを公開する考え方がなく、作業完了の報告だけでした。また10年以上前はweb上で自分の会社の情報を管理するという考え方があまり一般的ではなかった為、社内ローカルサーバーで情報を管理する事を前提にシステムも設計されていました。

4.計画した変更内容

部門を超え、全ての情報を一箇所から辿れるようにすること、状態の見える化を目標としました。

課題 1) ドキュメントのフォーマット

Text ファイル、Word ファイル、Excel ファイル、内容も部門、プロジェクトによりバラバラ

<対策>

全て JIRA 及び Confluence に統一。入力項目を制限することにより内容、精度、粒度も統一

課題 2) ドキュメントの管理

社内共有サーバーで管理、個人で管理

<対策>

全て JIRA 及び Confluence に統一。ファイルサイズにより共有サーバーで管理する必要があるものはどこにあるのかを記載。

課題 3) 部門間調整、情報共有方法がバラバラ

Excel やメールでの課題管理、課題の確認やり取り、複数の異なるシステムでの管理。

<対策>

全て JIRA に統一。JIRA のコメント機能を使い、メールやり取りをしなければいけない物も必ず JIRA に記録を残す。

課題 4) 部門間調整会議、ステークホルダーが多すぎる

複数の部門責任者が参加する会議で承認がないと決定ができない。

結果的に、関連する会議に全て出なければいけない

<対策>

JIRA でのフローを整備し、今まで会議で決定していた事項も JIRA でのフロー遷移やコメントのやり取りで完結させる。

課題 5) 使用システムが統一されていないため、情報を探すのが困難

複数の課題管理システムが使用されており、システムを超えての情報検索に時間がかかる

<対策>

全て JIRA 及び Confluence に統一し、トレーサビリティを確保する

5. 変更の実現方法

基板として JIRA を使い、過去のデータを可能な限り JIRA 上に移行しました。

それぞれ自分たちの都合の良いように作っていたものですので、使用している社内ユーザーからは当然今までのやり方と変わる事に対する抵抗があります。その抵抗を少なくするために、入力するシステムだけを変更しました。以前はテキストファイルや Excel でしたが、ファイルに入力していたものを Web システムに入力する「だけ」で他に新たに入力者に負担が増えない事を強調した結果、ほぼ抵抗なく全作業者に受け入れられました。部門により入力項目が異なるものも、目的や役割に合わせ統廃合しました。

また、JIRA でのフローも整備し、部門間でのギスギスしたやりとりを全て JIRA 上で状態遷移として見えるようにしました。

6. 変更後の状態や改善効果

情報を登録する作業は改善前とほぼ変わりません。

全ての情報を同じシステム上に載せ、フローを整備する事により、それまで多く発生していた部門間でのやりとりを整理する事ができ、会議も必要最低限のもののみを残し、廃止、縮小できました。目的の文章を探す時に、grep でネットワークフォルダを全検索していましたが、JIRA システムを使い一気に検索が可能になりました。誰が最新版を持っているのか、どこに必要な情報があるのかという問題も解決することができました。

対象	変更前	変更後	効果
仕様書	Text や Word をメールやり取り	Confluence、JIRA	• 全てのプロジェクトでフォーマットの統一 • 情報検索性の向上、トレーサビリティ確保
修正報告書	Text をメールやり取り	JIRA	• 状態の見える化による問題解決までの時間短縮
要望管理	システム A 及び Excel をメールやり取り	JIRA	• 責任の所在が明確 • 必要なステークホルダーだけ関与できる
ユーザー問い合わせ対応	システム B 及びメールリングリスト	JIRA	• 打ち合わせ工数削減
障害対応	システム C 及びメールやり取り	JIRA	
ドキュメントレビュー	Word、Excel をメールやり取り	JIRA	

Fig.2 : BeforAfter

7. 改善活動の妥当性確認

費用対効果という意味では既存社内システムとして構築されていた JIRA を使いました。具体的な会議工数の削減としては週 4 時間 20 人規模で行っていた部門間調整会議が週 30 分、5 人の会議に削減できました。

また、記録として残すべき情報はメールではなく、JIRA 等に残すよう意識改革が進んでいます。

課題としては、作ったシステム、プロセスがきちんと運用され続けるよう管理、運営をし続ける必要があります。また、組織の変化があった場合はそれに合わせ修正し続ける必要もあります。

2C1 「システム構築プロセスの標準化と CCSF 活用による IT 人材育成への取り組み」 清水淳史(セイコーエプソン)

<タイトル> システム構築プロセスの標準化と CCSF 活用による IT 人材育成の取り組み

<サブタイトル>

<発表者>

氏名（ふりがな）：清水 淳史（しみず あつし）

所属：セイコーエプソン株式会社

<共同執筆者>

氏名（ふりがな）：田中 秀樹（たなか ひでき）

所属：セイコーエプソン株式会社

<要旨>

当社の IT 人材育成活動の経緯と取り組みについて説明する。
過去の活動で一度失敗を経験しているが、その反省や気づきから新たな IT 人材育成モデルを設計し、仕組みを構築した。本発表では、IT 人材育成のためのシステム構築プロセス標準化と、CCSF（i コンピテンシ デイクショナリ）の導入と活用について説明し、今後の取り組みを紹介する。

<キーワード>

IT 人材育成、システム構築プロセス標準化、CCSF、i コンピテンシ デイクショナリ、組織力強化

<想定する聴衆>

組織長の方、組織の SEPG、IT 人材育成担当者

<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

私の所属部門でもある IT 推進本部は、本社組織として、IT による全社の情報基盤整備をミッションとして
いる。その中の社内向け IT を担当する情報化推進部において、2003 年度から IT 人材育成活動を企画
し、育成課題に対してスキルレベルの可視化とキャリア開発の促進を目指した。

2.改善前の状態

各自のスキルレベルの把握や部門の人材育成の仕組み整備に対する課題認識から、以下の仕組みを構
築し、2004 年度から 3 回の ITSS スキル診断を実施したが、目標設定が曖昧となり活動が停滞した。

- ・個人のキャリア開発およびキャリアの幅を広げる仕組みの整備（教育・研修体系、OJT 等）
- ・人材育成マネジメントサイクルの仕組みの運用

3.改善前の状態をもたらした原因（因果関係）

ITSS スキルレベルの把握が目的となってしまう、必要スキルやそのレベルが業務タスクと紐付けできていなか
った。また、ITSS をユーザ企業の情報システム部門に適用する上での機能不足もあった。
一方、2008 年のリーマンショックで当社も業績悪化に陥り、これまで外部ベンダーに頼っていたシステム発注
を外部流出コスト削減に向けて社員が自らシステム構築するよう経営側から内部化へのシフトが求められた。
この方針を受けた時に、限られた内部要員で組織機能すべてを実現できないという問題も見えてきた。これ
はシステム構築プロセスの標準化や導入技術、IT 製品の統制が不十分だったためである。

4.計画した変更内容

こうした要因分析から以下の 3 つの打ち手を考えた。

- 1.業務分掌の再定義を UISS を基に作成する
- 2.業務タスクとスキルの紐付けを CCSF ベースに行う
- 3.システム構築プロセスの標準化を行い、業務タスクを標準化する

これらを組み合わせ、IT 人材育成モデルを構築し、運用することとした。

5.変更の実現方法

IT 人材育成モデルでは、システム構築プロセスを標準化し、社員を計画的に育成するモデルとした。

1. システム構築プロセスを標準化し、そのタスク、デザインレビュー、各ステップでの成果物の明確化
2. プロセス定義から明確となる業務タスク、役割、教育コンテンツの定義と紐付け、ヒューマンスキルにおいて
は行動特性、キャリアパスの明確化
3. 中期計画を受けてその遂行に求められる組織機能である業務分掌、その組織機能を実現するために
求められるタスク・スキルの定義、それを基に目指すべき組織と個人の役割の明確化
4. 組織力強化のための PDCA、個人の自己成長支援のための PDCA サイクルを仕組みとして整備

6.変更後の状態や改善効果

CCSFで定める標準的な導入ステップに基づいて2013年度、約1年かけて仕組みの構築を行った。また、個人のスキル診断の環境構築のためにスキル診断ツールを導入し、スキル診断の環境を整えた。

目的としては、情報化中期戦略実現のために当社に必要な人材を育成すること、すなわち、IT戦略実現に向けて組織のミッションの達成とスキル継承が可能な強い組織運営をすることである。

そのための施策として、組織を軸とした組織力強化のための施策4つと、個人を軸とした自己成長支援のための施策2つ、合計6つの施策を実行した。いずれの施策もその基本となるのが、メンバー一人ひとりのスキルを棚卸し、組織と個人の実力を「見える化」することである。

各施策により、組織と個人の双方にIT人材育成の改善効果が見え始めてきている。

7.改善活動の妥当性確認

システム構築プロセスの標準化、およびCCSF(iコンピテンシディクショナリ)の導入と活用をベースとしたIT人材育成活動の取り組みにより、組織力強化と個人の自己成長支援への有効性が確認された。

今後は、更なる分析・利活用による運用の定着化を図るとともに、関連会社や本部全体、海外現法への水平展開を図っていく予定である。

2C2 「SQuBOK」と「ワークショップ」を活用した SQA 育成」菅沼由美子(パナソニック)

<タイトル>

「SQuBOK」と「ワークショップ」を活用した SQA 育成

<サブタイトル>

～ こんな SQA を育てよう！ ～

<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

<共同執筆者>

氏名（ふりがな）：

所属：

<要旨>

パナソニックの車載担当 SQA 育成のための、スキルの“物差し”である「SQA スキル要件」の座標軸に SQuBOK を活用することで、求めるスキルを漏れなく網羅的に示すことが出来た。また、テクニカルスキルと共に、ヒューマンスキル、プロセススキルを明示的にスキル要件に加えることで、求める SQA 像をより明確化することが出来た。更に、レベルの高い SQA 担当者の育成には、課題解決型のワークショップを用い、ワークショップデザインを含めた総合的な育成の仕組みを構築した。

<キーワード>

SQA、SQuBOK、スキル要件、育成、ワークショップ、テクニカルスキル、ヒューマンスキル、プロセススキル

<想定する聴衆>

SQA、SEPG、スキルマネジメントを担当される方、など

<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

筆者は、2010年から、パナソニックの電子部品カンパニー（部門）において、SQA 行政の一環として、SQA スキル要件を定め、カンパニー内の事業部 SQA 担当者の育成推進を行ってきた（既発表@SJ2010）。2013年、パナソニック横断の車載 SQA の育成を担当するにあたり、電子部品カンパニーの SQA スキル要件を、複数カンパニーにまたがるパナソニック全社車載向けに拡張すると共に、求める SQA 育成の仕組み構築に取り組んだ。

2. 改善前の状態

従来、電子部品カンパニーの SQA 育成のために用いていたスキル要件を、パナソニック全社で車載製品開発を担当する複数カンパニーの SQA 向けに拡張するにあたり、下記の課題が発生した。

- ・課題① 網羅的で汎用性のあるスキル要件とする
- ・課題② スキルアップのイメージを明確にする
- ・課題③ テクニカル以外のスキル要件を明確にする（レベル 4,5）

3. 改善前の状態をもたらした原因（因果関係）

電子部品カンパニーに限定された「SQA スキル要件」から、パナソニック横断の、複数カンパニーを対象とする「車載 SQA スキル要件」に拡張するにあたり、下記の変化点に対応する必要があった。

- ・変化点① ソフトウェアの規模と開発標準は、電子部品カンパニー内では、事業部が異なってもほぼ同様
→ パナソニック横断の、複数カンパニー対象の車載製品では、ソフトウェアの規模と開発標準が多様。
- ・変化点② 監査スキルのレベル感として、監査対象プロジェクトの種別（民生製品、車載製品など）により表現
→ 車載 SQA スキル要件では、車載製品のみが監査対象。
- ・変化点③ 対象範囲が電子部品部門 1 つのみ
→ パナソニックの車載全体が対象になり、事業部・カンパニーを跨る複雑な問題への、対処スキルが必要。

4. 計画した変更内容

変化点、課題に対する対策として、下記 3 点の対策を計画した。

- ・対策① ソフトウェア品質知識体系(SQuBOK)を活用する。
- ・対策② SQuBOK の項目ごとに、各レベルの必要スキルを定義する。
- ・対策③ ヒューマンスキル、プロセススキルを定義する。更に、ヒューマンスキル、プロセススキルの育成と確認のために、ワークショップを開催する。

5. 変更の実現方法

- ・SQuBOK は「カテゴリ」「知識領域」という階層構造を持っている。これを活かして、スキル要件の階層化を行う。
- ・SQuBOK の項目ごとに、SQA レベルに応じてスキルレベルが高くなることで、成長のイメージを明示する。
- ・SQA レベル 4、5 は、テクニカルスキルに加え、ヒューマンスキル、プロセススキルを明示的に規定する。そして、SQA レベル 4、5 に必要とされるスコープでの、課題抽出と対策立案を中心にしたワークショップを開設し、ヒューマンスキル、プロセススキル育成する。レベル 5 では、ワークショップの実施内容やスケジュールも、事務局ではなく、受講者が自ら決定・調整して実施する。

6. 変更後の状態や改善効果

「SQA スキル要件」への SQuBOK の応用により、網羅的で汎用性のあるスキル要件となり、スキルアップのイメージを明確にすることができた。また、レベル 4, 5 の SQA について、テクニカル以外のスキル要件を明確にし、育成のためのワークショップを行うことで、必要とするスキルを備えた SQA を育成する仕組みを構築することができた。

7. 改善活動の妥当性確認

本活動の結果として、目的とした改善に加え、スキル要件（テクニカルスキル以外も含め）を明確に定義することにより、SQA 育成のより系統立った実施と、スキル認定が行えるようになった。また、ワークショップで課題抽出と対策立案を行い、ワークショップ終了後に対策案を実施することで、広いスコープでの仕組み改善が継続して行えるようになった。ワークショップ終了後のスキル認定面談では、全受講者が合格し、求めるスキルが育成できることが確認できた。

2C3 「プロセス改善推進者の成長について」 中村伸裕(住友電気情報システム)

<タイトル>

プロセス改善推進者の成長について

<サブタイトル>

必要な能力とその獲得方法

<発表者>

氏名（ふりがな）： 中村 伸裕（なかむら のぶひろ）

所属：住友電気情報システム株式会社 QCD改善推進部

<共同執筆者>

氏名（ふりがな）：

所属：

<要旨>

プロセス改善推進者の育成は改善推進部門にとって重要なテーマであるが、必要な能力やその獲得方法が確立できていない。弊社ではプロセス改善推進者の成長を促進するため、これまでさまざまな取り組みを行ってきた。今回、プロセス改善推進者4名に対して個別に1時間程のインタビューを実施し、過去5年間のプロセス改善推進者の成長支援の活動を分析した。その結果を報告する。

<キーワード>

プロセス改善推進者、成長、モチベーション、能力、スキル、知識、自己実現

<想定する聴衆>

改善推進部門の管理者の方

<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（ ）

※事務局と相談させていただき、フリーフォーマットで投稿させていただきます。

<発表内容>

1. 概要

ソフトウェア開発組織では、入社直後から長期間にわたりソフトウェア開発に従事する技術者が多く、優秀な開発者が自然発生的に輩出されることも多い。しかし、改善推進部門のメンバーは開発部門から異動してくることが多く、プロセス改善で成果を得るためには、短期間でプロセス改善の能力を身に付ける必要がある。ところが、プロセス改善に必要な能力の定義やその獲得方法は確立されておらず、試行錯誤を繰り返している状態である。また、プロセス改善推進者の理想像は定まったものがなく、SPI Japan に参加されている実力者を見てもそれぞれ自分の強みを活かした個性的なスタイルになっている。今回、プロセス改善推進者 4 名に対して個別に 1 時間程のインタビューを実施し、過去 5 年間のプロセス改善推進者の成長支援の活動を分析した。

2. プロセス改善推進者に必要な能力の体系化

図 1 は筆者がプロセス改善推進者に必要な能力として想定したものを示している。プロセス改善は経験していない新しいプロセスを実装するという意味で非常に創造力を必要とする。取り組む内容も毎年変わるため、同じことを繰り返すスキルだけでは務まらない。プロセス改善推進者の最終形は “人生に究極の目標を定め、その実現のために努力する存在” であるマズローの自己実現の世界に入っていくものと考えた。組織のニーズに対して成果をあげるためには、プロセス改善の推進力が必要となるが、この能力を獲得する為には、プロセス改善の実績、スキル、専門能力が必要であり、さらにそのベースには人間としての行動特性があると考えた。プロセス改善の実績は、開発者からの敬意や上位者からの支持を得るために必要である。さらに実績を上げるためには、幅広い専門知識（ソフトウェア・エンジニアリングの知識）と改善活動のスキルが必要である。

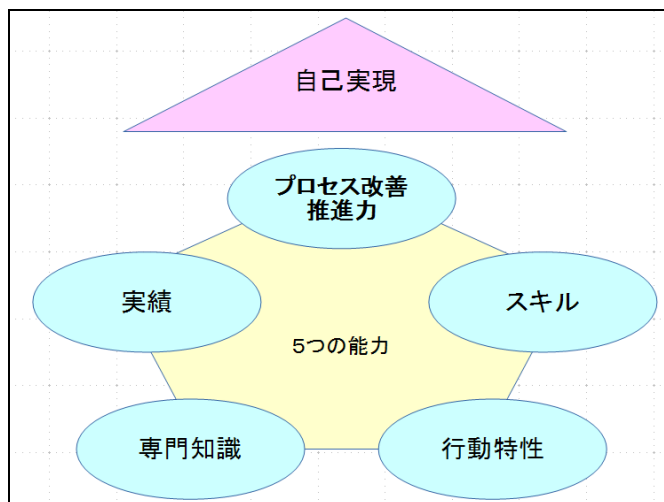


図 1. プロセス改善推進者に必要な能力

3. 人材育成の基本方針

プロセス改善推進者は “自ら考え、自ら行動する” 能力が必要であると考えている。この能力を身に付けるためには、人を育成するという本人からみて受け身の考え方ではなく、本人が主体的に試行錯誤しながら成長できる場を提供するという成長支援の考え方が適切である。図 1 の 5 つの能力のうち、プロセス改善の推進力は、他の 4 つの能力を総合的に

活用して発揮できる能力と考えている。また、行動特性については人間性の側面が強く、個別的に強化することが難しい。したがって、具体的な施策は、専門知識の強化、スキル強化、プロセス改善の実績を積み重ねる、の3点が対象となる。以下、4項で専門知識について、5項で実績とスキルについて説明する。6項では改善組織の文化が個人の行動特性に与える影響を説明する。

4. 専門知識の獲得

4.1 CMMI アプレイザルへの参加

(1) 狙い

組織内にプロセス改善をうまく定着させるためには、ソフトウェア・エンジニアリングの体系的な知識を開発部門と改善部門のメンバーで共有できている状態が望ましい。価値観と言葉が共有されていれば相互理解の工程が省略でき、価値の高い議論が可能になる。弊社ではソフトウェア・エンジニアリングの体系的な知識として CMMI のモデルを採用している。3年ごとに2回（2部門）公式アプレイザルを実施しており、アプレイザは毎回上限に近い10名を参加させている。また、公式アプレイザルの前のコンサルによる事前診断にも開発部門と改善部門から20～30名参加してもらい、繰り返しCMMIモデルの理解を深めてもらった。

(2) 評価

インタビューの結果、これらの活動に参加した改善推進者は全員、体系的な知識が得られたと感じており、改善活動のベースの知識として定着している。また、コミュニケーションの共通言語としても役に立っている。確実にプロセス改善推進者の知識を広げることができる取り組みである。

4.2 読書による専門知識の獲得

(1) 狙い

CMMIモデルは、ソフトウェア・エンジニアリングの全体像を把握するにはよいが、抽象度が高く、実際のプロセス改善ではより詳細な知識や具体的な事例が必要となる。ノウハウ的な知識はインターネットを検索すれば得られるものの、創造力を発揮するのに十分な知識の蓄積には書籍による学習が効果的であると考えている。施策としては、私自身が年間30～50冊の書籍を読み、読んだ本を部下の机に積み上げることで読書を促す手法をとった。

(2) 評価

この施策による効果は個人差があった。読書習慣が身についた改善推進者の動機は以下のものであった。

・わからないことがあって質問すると、詳しい答えが返ってくるが、知識不足でその答えが理解できない。

理解できるように書籍で勉強するようになった。

・知識不足により、同じ場にも見えている景色が異なる。同じ景色がみたい。

一方、読書習慣が身につかなかった改善推進者も書籍による知識獲得の重要性を感じていた。しかし、生活スタイルを変えるまでの影響力はなかった。今後の課題である。

4.3 コミュニティからの動向情報収集

(1) 狙い

社外から最新情報を得る能力を獲得してもらうために、SPI Japan, JUAS QCD 関西といった社外交流の場に積極的にプロセス改善推進者を参加させてきた。

(2) 評価

社外交流により他社でも同じような悩みを持っているという共感が得られたという意見が多いものの、積極的な情報収集のネットワークは十分に構築できていない。ただし、SPI Japan で発表のあった KWS 等の施策を社内に展開した事例はあり、社外の知識、ノウハウを社内に取り込むことはできている。社外からの情報収集は Facebook が効果的であると考えているが、Facebook による情報発信を制限している企業も多く、活用が難しい状況にある。

4.4 英語論文の説明会

(1) 狙い

社会人として求められている英語力の強化と欧米のソフトウェア・エンジニアリングの知識獲得を目指して、英語論文の説明会を定期的実施している。説明会は月に一度の頻度で開催し、各メンバーは年 2 回、改善推進メンバーに対して任意に選択した英語論文の内容を説明する制度である。以下の紹介された論文の例を示す。

- ・ Cem Kaner, J.D., "What Is a Good Test Case?"
- ・ Victor R. Basili, Gianluigi Caldiera, H. Dieter Rombach, "THE GOAL QUESTION METRIC APPROACH"
- ・ Laurie Williams, E. Michael Maximilien, Mladen Vouk, "Test-Driven Development as a Defect-Reduction Practice"
- ・ V. Basili et al., "GQM+Strategies in a Nutshell"

(2) 評価

当初、積極的ではないメンバーもいたが、実施してみると日本であまり知られていない知識が獲得できたり、英語の解釈を説明会で議論することで参加者の知識を統合する効果があったり、想定以上の成果をあげることができた。ただ、論文は 8 ページ程度であるため、知識の獲得量は少なく、これだけではプロセス改善推進者に必要な知識獲得としては不十分である。

5. 実績およびスキルの獲得

5.1 改善テーマの選定

(1) 狙い

弊社では目標管理の制度があり、毎年 4 月頃に年度計画を立案する。この計画はプロセス改善の成功実績を作るといった目的とプロセス改善推進のスキルを獲得するという目的で重要であると考えており、十分時間をとって計画している。これまでに取り上げた改善テーマの例を以下に示す。

- ・管理図の展開
- ・品質予測モデルの構築
- ・サービスサイエンスの導入
- ・アジャイルの評価
- ・テストデータ生成システムの開発
- ・GQM アプローチの展開

いずれも新規性が高く、世間や開発部門から注目を集めるテーマであり、開発部門の協力を得ながら主体的に進めることを期待して選定したテーマである。いずれも自ら考え、自ら行動できる環境を提供することが狙いである。

(2) 評価

インタビューの結果、設定したテーマに対する活動の結果はさまざまであった。インタビュー結果を以下に示す。

(a)期待した成果を得られなかったケースは以下のようなものであった。

- ・改善推進部門に異動してから日が浅く、改善推進者としての立ち位置がよくわからず、積極的に活動できなかった。
- ・システム開発部門へどんな貢献ができるか具体的にわからず、施策が打てなかった。

(b)成果が得られたテーマは以下のようなものであった。

- ・新規性が高く、取り組みに対するモチベーションが向上した
- ・開発部門からの期待が高く、モチベーションが向上した
- ・開発部門とうまく協力して進めることができた
- ・ツール開発では、自分が企画したものが実現でき、ツール開発によるプロセス改善の楽しさを知った。自信が付いた。

テーマへの取り組みは、本人の自発的な行動を期待しており、自走力のある人にとっては良い環境であるが、異動直後の推進者にとっては改善推進部門の立ち位置や具体的な進め方のアドバイスが不足していることがわかった。自分自身のアイデアで進めた取り組みが成功すれば、本人が自信を持ち、周囲からの評価が高まる効果が確認でき、テーマ選定は今後も力を入れていくべき価値の高い活動である。

5.2 SPI Japan での発表

(1) 狙い

SPI Japan での発表資料の作成は、自身の活動を社外からの視点で振り返るよい機会ととらえている。資料作成は品質を高めることを目的に提出期限ぎりぎりまで行っている。この課程でプロセス改善推進者が無意識に行っていた行動が形式知として論理的に説明できるようになったり、ものごとの因果関係が明確になったりする。また、社外発表により実施した改善活動の価値を組織内の管理職層や改善活動にかかわる担当者に伝え、適切な評価を得ることができるようになる。結果として、WGのメンバーをうまく巻き込んで改善活動を推進する能力の獲得を狙っている。

(2) 評価

インタビューの結果、SPI Japan で発表したことのあるプロセス改善推進者は社外でも発表できるという自信を獲得しており、積極的に改善活動を進めるモチベーションの基礎になっていると考えられる。SPI Japan での発表は業績報告会等で報告されており、組織内での評価にもつながっている。

6. 行動特性

行動特性は、個人の素質によるところが大きく、研修等による手段で向上させることは難しい。しかし、改善推進部門の活動がアクティブであれば、その影響を受け、個人の行動特性が変化していることが確認できた。以下、インタビュー結果から価値観、行動特性の変化に該当するものを示す。

- ・以前は品質改善といえはチェックリスト位しか思いつかなかったが、現在はプロセスを考える
- ・CMMI の知識が得られたことで、プロセスの不足事項に気がつくようになった
- ・プロセス間の関係を理解することで、より根本的な原因が見えるようになった
- ・品質が悪いとコスト、納期も悪化することを実感
- ・新しいことに積極的に取り組むようになった
- ・事業部から要求されたことをするのではなく、自ら企画するようになった
- ・長期間で物事が考えられるようになった

7. 開発部門からの評価

改善推進部門の能力を評価するために開発部門の部長、課長、係長相当の役職の管理職 8 名にアンケート調査を実施した。その結果、CMMI の評定能力、実績データの収集・分析能力、メトリクスの定義能力等で高い評価が得られた。CMMI モデルをベースとした能力が獲得できていることが確認できる。一方、開発者が持つ暗黙知をソフトウェア・エンジニアリングの知識をベースに形式知化・モデル化する能力、世の中の動向を把握する能力、現場の実態を把握する能力等が今後期待する能力としてあげられた。改善推進部門全体として知識不足が指摘される結果となった。4.2、4.3 項の課題がはっきりと示されている。

8. 改善推進者の能力を引き上げる指導者像

リーダーの性質は、増幅型リーダーと消耗型リーダーの間はどこかにあるという考え方がある。消耗型リーダーは厳しい管理体制を敷き、チームの創造力を十分生かしきれないリーダーで人材育成よりも成果を重視するタイプである。一方、増幅型リーダーは以下の性質を持つ。

- ・チームの人々の持つ能力を見抜き、発揮させる
- ・周囲の人々を育て、最高の力を発揮させる
- ・やりがいのある目標を定め、従業員にできると思っていなかったような劇的な成果を生み出す

人材育成という観点では増幅型リーダーの性質をより多く持つことが期待される。人材育成と成果の両立を追求する上で考慮すべき事項だと思われる。

9. 結論

プロセス改善推進者に必要な能力として図 1 を想定して、成長支援の取り組みを進めてきた。インタビューの結果からは筆者の意図通りに進んだものと、そうでなかったものがあった。

プロセス改善に必要なスキルとプロセス改善の実績に関しては改善点があるものの目標管理制度と連携して進めることで一定の成果があげられることが確認できた。行動特性は改善推進部門の文化によるところが大きく、具体的な施策がなくてもよいことが確認できた。

プロセス改善に必要な知識の獲得については CMMI のアプレイザル活動を通じて、ソフトウェア・エンジニアリングの基礎が構築できることが確認できた。また、英語論文の説明会による知識獲得にも期待が持てる。しかし、多くの改善推進者が専門知識の不足を強く感じているにもかかわらず、実践に必要なより具体的な知識や幅の広い知識を書籍から獲得することに関しては習慣化できない改善推進者も存在している。毎年、異なるテーマで改善を進めるプロセス改善者にとって、これで十分という知識量はなく、継続的に知識を獲得する能力は今後の課題である。

以上

2C4 「機能安全開発を支える安全文化の確立と実践」 安倍秀二(パナソニック)

<タイトル>

機能安全開発を支える安全文化の確立と実践

<サブタイトル>

～機能安全開発スキル強化とスキル認定制度～

<発表者>

氏名（ふりがな）：安倍秀二（あべしゅうじ）

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

<共同執筆者>

氏名（ふりがな）：野吾英俊（やごひでとし）

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

<要旨>

2011年に自動車用の機能安全規格 ISO 26262 が制定され、国内外のカーメーカ及び部品メーカからの機能安全開発要求が急増している。開発現場としては、機能安全開発のスキルを有する技術者、品質担当者の育成が急務となった。2012年からセミナー提供を開始したが、体系的なセミナー構成となっておらず、実施する講師も不十分であった。2014年からはスキル認定制度を開始し、体系的なセミナー提供と目標スキル到達とセミナー受講との関連づけをした。また、セミナー講師及び現場の機能安全推進のキーマン育成のしくみも試行した。2014年度の実績は、セミナ提供数*回、のべ受講者数*名となった。

<キーワード>

スキル強化、スキル認定、機能安全、機能安全開発、安全文化、セミナー、指導者育成、コンサルティング

<想定する聴衆>

教育担当者、SEPG、エンジニア、SQA

<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

2011年に自動車用の機能安全規格 ISO 26262 が制定され、国内外のカーメーカ及び部品メーカからの機能安全開発要求が急増している。開発現場としては、機能安全開発のスキルを有する技術者、品質担当者の育成が急務となった。組織においては、スキル保有者の見える化や計画的な育成、受講者には、受講のモチベーションアップが必要。セミナーの提供側としては、セミナー講師のスキル強化、講師数の拡充が必要である。また、受講者に対しては、セミナー提供だけでは、知識を得るだけであり、開発スキル強化としては弱いため、現場の開発支援を行う機能安全コンサルタント及び開発現場の指導者の育成も併せて行う必要があった。

2.改善前の状態

2012、2013年とセミナー提供を実施したが、セミナーが体系的になっておらず、希望者にセミナー提供を実施するのみであった。セミナー受講に関する組織の関与も弱く、必ずしも計画的な実施になっていなかった。セミナーを実施する講師も当初は*人しかおらず、2年目も*名体制で実施したが不十分であった。これらを解決するために、次の内容を実施することとした。

- スキルのランク付けと見える化
- 体系的なセミナー構成とスキル提供
- 受講に対する受講者のモチベーションアップ
- セミナー講師の拡充
- 機能安全コンサルタントの育成

3.改善前の状態をもたらした原因（因果関係）

開発現場のマネジメントを巻き込んだ活動になっておらず、また、受講者の見える化も出来ておらず、組織での計画的な育成に至っていないかった。セミナー受講後は、受けっぱなしになっており、適切なフォローもできていなかった。講師スキルは、育成の観点がなく、とにかくできるものが担当するといった状態であった。

上記の内容を適切に実施するプロセスが不十分であったことも大きな原因である。

4.計画した変更内容

2014年に改善を計画した項目は以下になる。

- 組織の計画的な育成とするためにスキル認定制度を制定する。
- セミナーテキストを再構成し、体系的なセミナー提供が出来るように改善する。
- セミナー講師不足が予測されたので、講師育成のしくみも組み込む。
- 知識提供のセミナー実施に加えて、スキル強化の活動も実施する。

5.変更の実現方法

(1)スキル強化のしくみ制定

まず、スキル強化を実施するために、スキル認定ガイドラインを策定し発行した。本ガイドラインの内容は次のとおりである。

- スキルレベル定義
- スキル認定方法
- スキルマップ
- セミナー体系

スキルレベルを5段階に設定し、ブロンズ、シルバー、ゴールド、プラチナ、ダイヤモンドとし、それぞれのランクに求められる内容を明確にした。また、開発におけるロールをひもつけた。各ランクに達成するために受講すべきセミナーを明確にし、スキルの認定方法も定義した。

(2)セミナー体系

提供するセミナーと内容を表 1 に示す。

コース名称	内容
機能安全実用コース導入編	機能安全規格 ISO26262 の Part2 (安全管理) を中心とした機能安全の備え、開発の概要
機能安全実用コースシステム編	機能安全規格 ISO26262 の Part3 (コンセプトフェーズ) ,Part4(システム)を中心とした、要求分析やアーキテクチャ設計、システムテスト
機能安全実用コースハードウェア編	機能安全規格 ISO26262 の Part 5 (ハードウェア開発) を中心とした、ハードウェア設計、安全分析、メトリクス算出
機能安全実用コースソフトウェア編	機能安全規格 ISO26262 の Part6 (ソフトウェア開発) を中心とした、SW 要求分析や SW アーキテクチャ設計、ユニット設計、SW テスト
システム製品開発プロセス 実用コース	機能安全対応標準プロセスの紹介と実施上の留意点
Automotive SPICE 入門コース	AutomotiveSPICE®の各プラクティスの解説

(3)スキル認定

スキル認定については、スキル認定ガイドラインに従って実施し、達成者には認定証を発行し、所属する組織の責任者から授与を行うこと、全社のスキル認定の登録を行うことにより、受講完了者のモチベーションもアップした。組織では、マネジメントの関与も進め、スキル達成者の管理を行うことにより、計画的な受講も実現しつつある。

(4)エキスパート育成

セミナー実施だけでは、スキルアップは不十分で、スキルランク 3 のゴールドレベル達成のために、(2)のセミナーとは別の“機能安全ワークショップ”を実施した。ゴールドレベルの人材は、各部門の開発現場の機能安全開発推進のリーダーとなることを目的としており、各部門から*~*名を選抜し、6ヶ月 12 日間のワークショップを受講する。ワークショップでは、ISO26262 規格の解釈、演習、ディスカッション、事例交流などを行う。これにより、真のスキルアップを狙った。

(4)機能安全セミナー講師、コンサルタントの育成

我々、機能安全推進部門では、セミナー運営、講師提供、現場の開発コンサルティングを担当している。これらを推進することができるスキルを向上させるのは、講師を担当することが早道である。講師を務めるためには、機能安全規格の字面を理解するだけでは不十分で、その行間や周辺の規格や情報との関係、業界動向、実開発との関連などを理解する必要がある。そのために、セミナーに参加し、先輩講師のやり方などを見せることと、講師自らがテキストを作成し、講師からの直接的な指導などを実施した。

6. 変更後の状態や改善効果

2014 年の活動を経て、実施前に計画した目標はほぼ達成できた。定量的には、表 2 に示すように

	2013 年度	2014 年度	2013⇒2014
セミナー実施	*回	*回	3.2 倍
参加人数	*名	*名	2.6 倍
スキル認定数	*名	*名	2.4 倍
機能安全エキスパート数	*名	*名	3.3 倍
セミナー講師	*名	*名	3.3 倍

2013 年度のスキル認定は、制度がなかったため、スキル認定制度開始後に見なして認定した。

これらを実現できた理由は

- スキル認定ガイドラインによるスキル基準を設定したため、セミナー参加も目標が明確になった。
- 体系的なセミナーとしたため、受講が容易になった。
- スキル認定者数を管理することにより、組織ごとの計画的なスキル強化が可能になった。
- スキル認定を組織に認められることによる受講のモチベーションアップにつながった。
- 講師育成により、より多くのセミナー提供が可能になり、現場のニーズに応えられた。
- 機能安全ワークショップによって、参加者による議論により知識も深めることが出た。また、開発部門を超えた横連携ができ、より実質的な情報交換を直接に実施できるという副次効果も得られた。

7. 改善活動の妥当性確認

2014 年の課題としては、即戦力という点ではセミナーだけでは弱く、加えて、機能安全ワークショップで実施した達成度テストの分析から、現場を指導するという点では、まだまだ、合格には達成していないということがわかった。これらを埋めるために、2015 年度は、現場の困り毎を中心に、ハンズオンセミナーを実施する。また、それらをトレーニング材料として蓄積し、組織横断的に活用することにより、より効率的なスキル強化を実施する。セミナー講師が増えたことにより、講師への負担が減り、こういったアプローチもより強化できる。また、機能安全ワークショップについては、2014 年度に、平行して作成した教材を、本年度は、事前に渡し、予習が出来るようにすることと、より実践的になるよう自分で考え、議論ができるような演習を増やす予定である（6 月後半から開始予定）。そして、新たに、機能安全エキスパートのさらなるスキル強化と交流の場を提供するために、機能安全コミュニティを新設し、年 4 回実施する。現在の**名のエキスパートを対象に、現場の実開発での困りごとを取り上げ、議論を行い、参加者で共有する開始する予定である。これらのように、2015 年度は、2014 年度の課題を解決するためにさらに改善をして行く予定である。

※本文中の具体的な数字については、伏せ字にしているが、当日のプレゼンでは掲載予定である。

3A1 「会社が用意したプロジェクト管理ツールを私が使いこなせている 2 つの要素」 高橋裕之(ウイングアーク 1st)

<タイトル>

会社が用意したプロジェクト管理ツールを私が使いこなせている 2 つの要素

<サブタイトル>

～私にください、管理者権限を～

<発表者>

氏名（ふりがな）： 高橋 裕之（たかはし ひろゆき）

所属：ウイングアーク 1 s t 株式会社

<共同執筆者>

氏名（ふりがな）：

所属：

<要旨>

現在のソフトウェア開発は短納期・高品質を強く要求されている。そこでタスク、課題、リスク、進捗などを効率よく管理し、かつ組織全体の情報共有を実現するためには何らかのプロジェクト管理ツールが必要不可欠である。

弊社では「開発共通インフラ」の一つとして JIRA が導入されたが初めのうちはあまり利用されていなかった。しかし、今では複数の製品化プロセスの様々な部分を担っており効果的に活用出来ている。プロジェクト管理ツール導入を成功させるために必要と考える 2 つの要素「SPI リテラシー」と「管理者権限」について論じてみたい。

<キーワード>

SPI、PMO、SEPG、ツール、JIRA、Confluence、プロジェクト管理、タスク、課題、リスク

<想定する聴衆>

ソフトウェアエンジニア、PMO、SEPG、SQA、情シス、プロセス改善担当者

<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

近年の開発担当者にとって、プロジェクト管理ツールを一から選定し導入に漕ぎ着けることは難しくなっている。

一昔まえであれば、勝手にサーバーを建ててオープンソース（OSS）の Trac や Redmine をインストールし、サクッと運用を開始して管理も上手く出来ていたであろう。だが、今やOSSはセキュリティの面のリスクが高く個人では責任が取れない（2014年の OpenSSL 脆弱性問題は記憶に新しい）。サーバーのハードウェアリソースは VM 化やクラウド化が進み、情報システム部門（情シス）を通さずに何かをすることは費用面、運用面、バックアップなどの保守面など様々な観点から困難になっている。

古き良き「勝手サーバー」に関しても、しばらくするとサーバー管理者の負荷が上がり（もしくは担当者がチームを外れたり、偉くなったりで…）メンテナンスそのものが誰も出来なくなってしまうケースは良く聞く話である。

このような背景からソフトウェア開発組織はプロジェクト管理ツールやソースファイルバージョン管理ツールなどのリソースを集約し「開発共通インフラ」や「情報共有基盤」と呼ばれる全社共通環境を構築し、運営は情シスの責任とし提供することで利用の効率化を図る。これは必然的な流れであろう。

ウイングアーク1 s t 株式会社は、いくつかのソフトウェアベンダが合併して出来た企業ということもあって、開発チーム毎にプロジェクト管理ツールが乱立していた経緯がある。そこで情シス主体で「開発共通インフラ」の整備を進め、課題管理とコンテンツ管理に JIRA（プロジェクト管理ツール）と Confluence（情報共有ツール）の導入が決定した。

これは開発担当者側から見れば「会社が用意したツール」を、どうにか使いこなす必要に迫られた状況とも言えた。

2.改善前の状態

程なく JIRA や Confluence が利用可能となったが、開発プロジェクトでは主に3つのパターンが発生していた。

- ① 使おうとしない。
- ② 使ってみたが何かがおかしい。
 - タスクチケットのステータスが Open のまま。タスクはもう終わっているのに。
 - JIRA に課題チケットがどんどん溜まる一方で少しも減らない。
 - チケットを見てもどんな作業かわからない。必要なことが書かれていない。
 - 激しいコメントの応酬があるのに、担当者はずっと同じ人。
- ③ 積極的に使いたいのに、何かもどかしい
 - 画面の項目を増やしたいが、誰に言えばいいのかわからない。
 - JIRA ワークフローがデフォルトのまま運営されている。
 - レガシー（既存）システムを JIRA 置き換えたいが、何をどうやって始めればいいのかわからない。
 - Confluence が自由すぎて、ドキュメントが読みにくい、可視性が悪い。

3.改善前の状態をもたらした原因（因果関係）

有識者によるツール選定プロセスを経て導入されたシステムであるが「さあ！使いなさい！」と言われただけ

では現場は戸惑うだけである。また、トレーニングをしっかりと実施したとしても、それはツールの知識が得られるだけで使いこなせるとは限らない。

ツールを「使おうとしない」という状態は、そこにある「プロジェクト管理ツール」と現場にいる自分たちの課題との関連性を見いだせないからだろう。

ところが、これがもしチームが独断でサーバーを立てて Redmine をインストールし利用し始めていたなら、トレーニングなど受けずとも Web や書籍で必死に情報を集め、がむしゃらになって運用をしていた事だろう。（きっと、デブサミで発表しちゃったりもするだろう）

この差はなんだろうか？

現場の人間が能動的に動くとき、それは現場の問題（課題）の認識がきちんと出来ていて、それを解決するために必然的な行動をしたに過ぎない。結果が伴うか（上手く問題が解決するか）はまた別の話なのだが、少なくとも行動を始めたときの「ゴール」は見失っていないはずだ。

SPI 活動に関わる人にとっては当たり前であるが、改善はまずは状況の分析から初め、現場の人間が課題を認識していなかったら、それを認識させることから始める。プロセス改善の本質を理解しているので、限られたリソースのなかで優先順位を付けながら改善計画をたて IDEAL モデルを回しているはずで、「さあ！ここにあるツールを使って！」とだけ言っているのであれば、いきなり行動フェーズ（A）から初めているようなものであり、上手いかないのは自明である。

この点に気づくか気付かないかの差を「SPI リテラシー」の差と名付け、SPI リテラシーが高いエンジニアや SEPG、PMO の存在如何で、その後プロジェクト管理ツールの活用には大きな差が生まれると考えている。

4.計画した変更内容

まず、運営体制に課題があると考えた。そもそも、プロジェクト管理ツールにおける導入から実運営までの「やらなければならないタスク」を、「情シス」と「現場」の2つの組織だけでやろうとするから無理が出るのである。そこで、これを3つのドメインに分けて捉えるようにする。（図1）

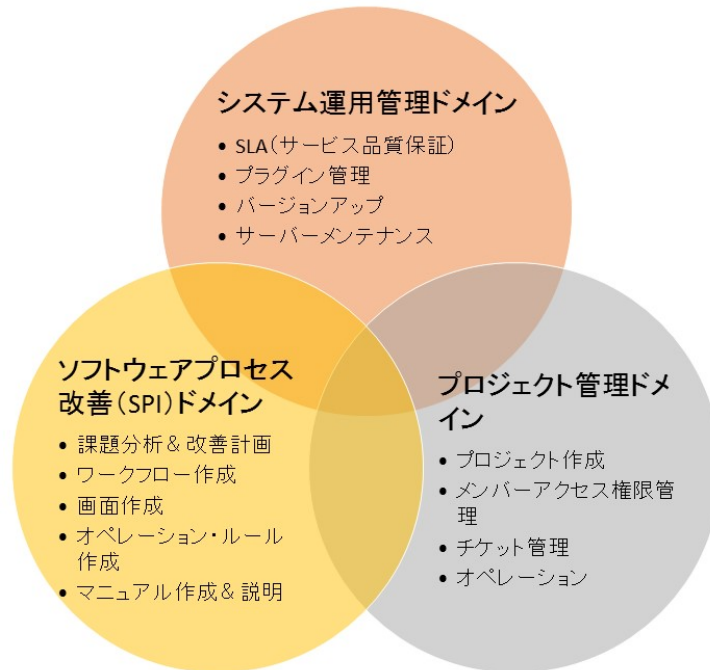


図 2 プロジェクト管理ツールの運営領域

上の図で記した通り、「システム運用管理ドメイン」は従来通り「情シス」の責務である。また当然ながら「プロジェクト管理ドメイン」は「現場」の責務だ。しかし、「ソフトウェアプロセス改善ドメイン」に書いた各タスクは、実は専門性が高く、ここをプロセス改善の専門家である私のチームが責任を持つべきと考えた。

こうして3つのドメインで役割分担を明確にしつつ、三位一体のツール運営体制構築を計画した。

5. 変更の実現方法

A) 3つのドメイン体制構築

まずオペレーションコストが高く課題も多かった既存システムの一つを、JIRAへ置き換える作業から開始した。この過程で現状分析とJIRAシステム移管に関するFit&Gap分析し、情報システム部門に相談しながら作業を行うことで、自然と（しかし意図を持って）3つのドメイン体制にした。

ここで重要なのは、計画を立ててそれを見せながら「行き当たりばったり」ではないとステークホルダーを安心させながら行動することである。そして、情報システム部門からの信頼を得てプロジェクト管理システムの管理者権限を付与して頂くことが出来た。

また、開発部門のトップからも活動の理解を得ることが出来、その後の施策が実施しやすくなった。

B) 課題分析&改善計画

計画されないものは実行されない。現場で明らかに横たわる課題から「ささやかれる」不満まですべてを分析対象とし改善計画を立てるようにした。

また、計画をステークホルダーにレビューしてもらうことで見逃していた課題が見つかることがあった。

C) ワークフロー作成

JIRA でデフォルトとして用意されているワークフローは役に立たない。何故なら、ステータスからステータスの状態遷移が何処からでも出来てしまし「自由度」が高すぎるのである。絵にすると形は綺麗だが、各ステータスからクローズ（Closed）への遷移が可能である。これは意味があるのだろうか？（図 2）

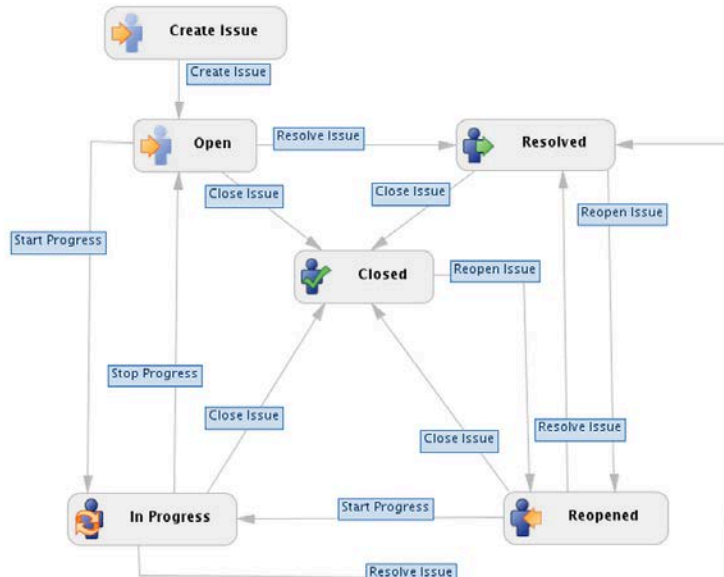


図 2 JIRA のデフォルトワークフロー

経済行動学に「決定回避の法則」というものがある（提唱者：Eldar Shafir）。「人は選択肢が多くなると逆に行動を起こせなくなる」というものだが、ワークフローにも同様の事が言えると思う。デフォルトワークフローを運用すると、きちんと

- ・ **Open → In Progress → Resolved → Closed**

というライフサイクルを意識出来るひともいれば、

- ・ **Open → Closed**

と極端にショートカットする人も、必ず現れる。

ワークフローの設計は、「SPI リテラシー」が高ければソフトウェアエンジニアリングにヒントを得れば良いことを知っている。なんらかの仕事をさせるのが「ワークフロー」なのだから、それはすなわちコンピューターの OS と同じことである。

例えば μITRON4.0 のタスク状態遷移は洗練されていて無駄がなく美しい。

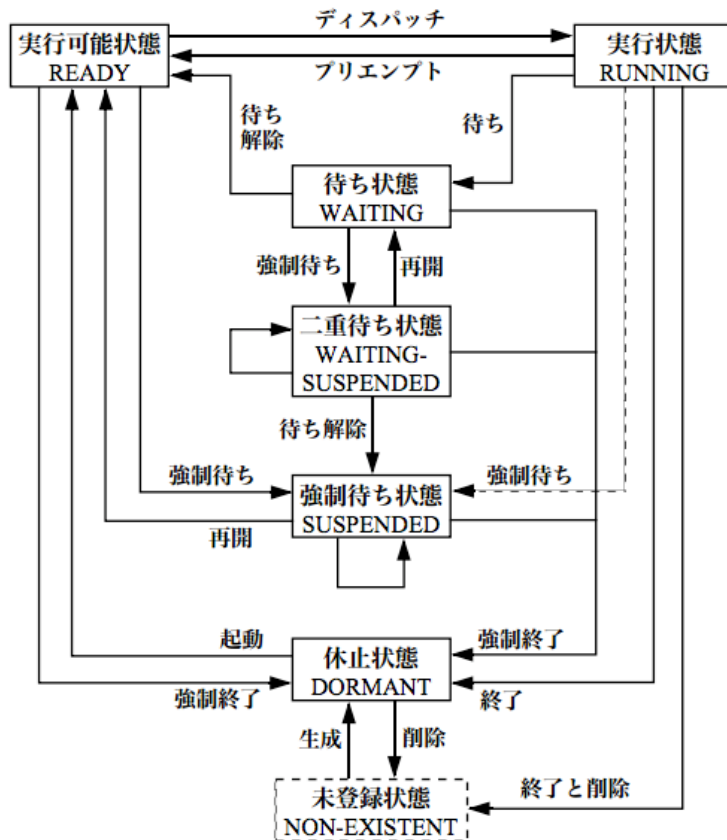


図 3 μITRON4.0 タスクの状態遷移

ワークフローとタスク状態遷移はとても良く似ている。「JIRA に何をさせたいのか？」という要件をもとに設計すれば、自ずと洗練されたワークフローを作ることができるのである。(図 4)

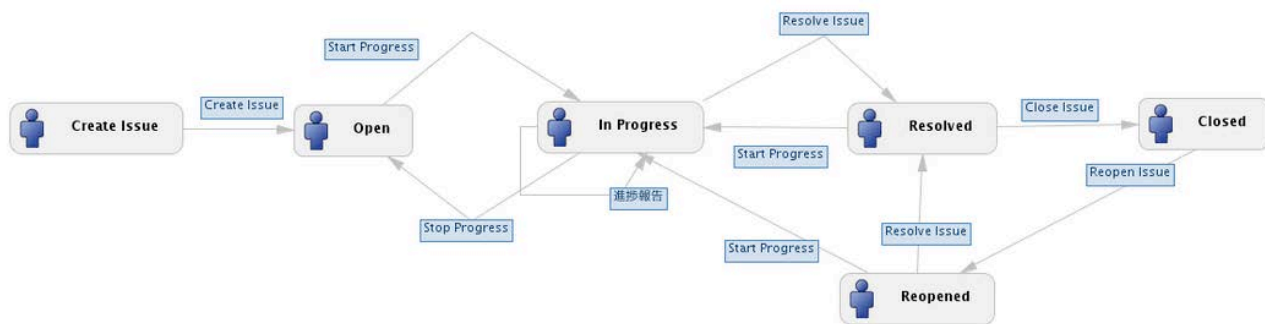


図 4 インプリメント済みの JIRA ワークフロー例

D) 画面作成

Web システムの利点である必須項目、オプション項目をきちんと考えた画面作りを行った。レガシーシステムでは「未記入」でもチケットを起票することが出来たので、運用中は情報不足により様々な問題が発生していたが、これを回避することが出来た。

E) オペレーション・ルール作成

実際にオペレーションするのは人である。これは SPI でも重要なテーマであるので、行う施策も同じである。まず役割分担をはっきりさせる。ステータス毎に関わる部門、ロールは何か、きちんと明文化して Confluence に記載した。また、チケットには「期日」という賞味期限があり、賞味期限が切れていないか定期的に棚卸しが必要となってくる。これは当然現場リーダーの仕事とした。

このように、当たり前である行動原則であっても、きちんとマニュアルに含めることで実際の行動につなげることが出来たと思う。もちろん、実際にオペレーションが回しているか？という点はプロセス QA のテクニックで、我々支援グループがプロセス監査を行うようにした。

F) マニュアル作成 & 説明

すべての情報は体系化し Confluence に記載、検索可能状態にした。あらゆる情報を Confluence に集約することで、「情報共有ツール」にどのようなメリットがあるかを体感させるように誘導した。また、JIRA プロジェクトをカットオーバーする際は、きちんと説明会を開催した。

この様に、これほどの施策を実施するには大前提として「JIRA の管理者権限を持っている」事がとても重要である。私のほかの企業での経験、また JIRA を利用するエンジニア仲間の話を聞くと、JIRA の管理者権限をごく一部の人間（主に情シス）にだけ集約している組織がとても多い。

しかし、そのような環境では SPI 活動の基本である IDEAL モデルは絶対に回らない。SPI リテラシーが高い人や部門が、ツールの管理者権限を得ることはツール活用の絶対条件である。

6. 変更後の状態や改善効果

新規開発プロジェクトのプロジェクトマネジメントに必ず JIRA が利用されるようになった。

現在は 23 個の JIRA プロジェクトが運用され、プロジェクトの特性毎に、WBS やかんばん、Scrum ボードを使い分けている。

また、組織間の課題解決にも 10 個の JIRA プロジェクトが活躍している。このうち 8 プロジェクトはレガシーシステムから JIRA への置き換えである。

7. 改善活動の妥当性確認

これから益々（有償／無償に係わらず）プロジェクト管理ツールの集約化と活用のプレッシャーは増えていくと思われる。その時、SPI の視点がないプロジェクト管理ツールの運用は、局所的な「管理者権限」集約を招き、非常に使いにくく、利用率の低いシステムになってしまうだろう。

これを回避するには「SPI リテラシー」の高い人員を配置し、そこにツールの「管理者権限」を付与することはプロジェクト管理ツール活用の絶対条件となっていると経験によって確信している。

3A2 「Rubyを「やりたい」から「やった」にするまでの道のり」 細美彰宏(日立ソリューションズ)

<タイトル>

Rubyを「やりたい」から「やった」にするまでの道のり

<サブタイトル>

<発表者>

氏名(ふりがな)：細美彰宏(さいみあきひろ)

所属：株式会社 日立ソリューションズ

<共同執筆者>

氏名(ふりがな)：牧俊男(まきとしお)

所属：株式会社 日立ソリューションズ

<要旨>

当社では 2008 年から Ruby を使った開発の取り組みを行っているが、エンタープライズ向けの大規模システムを Ruby で構築、カットオーバーさせる実績を作るまでの道のりは平坦なものではなかった。当社が Ruby を取り組み始めた当時から現在に至るまで、試行錯誤を繰り返し、どのように考え、施策を打って、社内において、使える開発技術として認知されるようになったかを紹介する。プロセス改善に対する取り組みを如何に開発現場に浸透させ、使ってもらうようにするか、そのヒントになればと思う。

<キーワード>

Ruby、活用推進、普及活動、認知度向上、技術整備

<想定する聴衆>

ソフトウェアエンジニア

<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

Ruby は日本発のオブジェクト指向スクリプト言語で、Ruby on Rails という Web アプリケーションフレームワークの登場により、Ruby に対する注目度が高まった。2007 年日経 BP 技術賞大賞にプログラミング言語 Ruby が選ばれ、社内で話題に上がったことが最初のキッカケ。当社では 2008 年から Ruby 開発の取り組みを始め、専門組織を設置して、Ruby の研究開発、技術整備、活用推進に取り組んだ。

2.改善前の状態

エンタープライズ向けのシステム構築で採用されるプログラミング言語は Java や .NET、COBOL が主流であり、Ruby はまだまだマイナー。社内に、Ruby に興味を持っており、「やりたい」エンジニアはいるが、開発事例やノウハウが少なく、実際にプロジェクトで使ってみようかと検討するエンジニアは少数で、「やった」ことのあるエンジニアは稀だった。Ruby 技術者・経験者が少ないため、活用に不安を感じる意見が多かった。

3.改善前の状態をもたらした原因（因果関係）

開発事例やノウハウが少ない、技術者・経験者が少ないということが課題。実際にプロジェクトを立ち上げようとした場合、開発手順や環境、管理指標などの技術的な課題、開発要員が集められないという技術者不足の課題があった。

4.計画した変更内容

Ruby を普及させるために何が必要かを考えて、少しずつ整備を進めた。開発環境、開発手順・規約、技術教育、問い合わせ対応などの整備を進めた。また、認知度向上のため、社内 PR も並行して行った。

5.変更の実現方法

2008 年、Ruby に知見のあったエンジニアを集め、Ruby の評価から始めた。Ruby を使って、社内コミュニティシステムを開発して、Ruby でいける、という実感を得たことで、本格的に社内に普及する活動を始動。2009 年 12 月に Ruby 専門組織を作り、Ruby の技術整備を進め、社内に普及させる取り組みを行ってきた。

- ・ 開発環境：設計からプログラミング、テストまでの開発が行える環境を構築済みの VM 提供。
- ・ 開発手順・規約：試行開発や早い段階での活用プロジェクトでのノウハウを手順に整備。
- ・ 技術教育：教育部門と協力して、技術教育コースを開講。Ruby ができるパートナー会社と関係構築。
- ・ 問い合わせ対応：Ruby 専門組織が問い合わせを受け、対応。

6.変更後の状態や改善効果

開発環境、開発手順・規約、技術教育の整備を進めながら、Ruby の活用推進を行ってきた。最初の活用推進のターゲットは、Java の開発では採算が合わない周辺システムや中小規模のシステム構築。数件の部署が積極的に活用するようになったが、ちょっとした小規模の開発にしか使われないという悩みがあった。小さいプロジェクトには適用できるが、ちょっと大きなプロジェクトだと、実績のない言語は使えない、リスクは取れないという反応だった。

当社の開発案件への適用には限界を感じ、行き詰っていたが、Java で開発されたシステムの保守の効率化を検討している大規模案件があり、ロビー活動により、Ruby を採用。更に、Java で開発すると規模が大きすぎて、期間内に納品できない案件があり、Ruby を提案。両プロジェクト共に、Ruby 専門組織の技術者が入り込んで、一緒にやってみせ、「やった」実績を作った。両システムのリリース稼働を間に合わせることができ、社内での認知度・評価が一気に上がった。同時に、社内の「やりたい」という声も増えた。

7.改善活動の妥当性確認

開発環境、開発手順・規約、技術教育、問い合わせ対応などの整備を行うだけでは、「やりたい」から「やった」にするのは限界があった。やってみせることで、「やった」に変わり、「やりたい」も増やすことができた。今後は、「やった」を「次もやりたい」というレポートを増やすこと。

3A3 「ウォーターフォールでスクラム Enjoy !」 松澤将明(富士通ソーシャルサイエンスラボラトリ)

<タイトル>

ウォーターフォールでスクラム Enjoy !

<サブタイトル>

ウォーターフォール型開発におけるスクラムでの自律したチーム創り

<発表者>

氏名(ふりがな) : 松澤 将明(まつざわ まさあき)

所属 : 株式会社富士通ソーシャルサイエンスラボラトリ 公共ビジネス本部 第一システム部

<共同執筆者>

氏名(ふりがな) :

所属 :

<要旨>

私が所属するプロジェクトは、全プロジェクトメンバー約 300 名以上、チーム数 20 を超える大規模プロジェクトであり、どのグループにおいてもウォーターフォール型を主とした開発を行っている。しかし、私のチームでは、ウォーターフォール型開発での改善プロセス・チーム創りに疑問を抱いたマネージャーがスクラムを導入している。

ウォーターフォール型開発に、アジャイルプラクティスを取り入れ 3 年が経過し、開発及び運用・保守を並行で進めるチームにおいて、各メンバが試行錯誤しながら自律したチームを目指す姿を紹介する。

<キーワード>

スクラム、ウォーターフォール型開発、大規模プロジェクト、自律したチーム、情報の共有、ふりかえり

<想定する聴衆>

ウォーターフォール型開発にスクラムを導入している方、スクラムを導入し運用しているリーダー
自律したチーム創りを目指している方

<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

担当チーム（10名～20名）におけるスクラムの導入目的は自律したチーム創りであり、自律したチームによりチームが開発に専念し、開発を楽しめる環境を期待した。しかし、導入当初は、自律したチーム創りとは程遠い状況であった。導入目的としている「自律したチーム」を創るために、取り組んだ活動の事例である。

2.改善前の状態

スクラムフレームワークを導入するだけでは、自律したチームは創られないと判断した。判断理由としてスクラムを運用するなかで問題となった「モチベーションの低下」「情報の共有不足」の2点が挙げられる。この2点を改善することで、より自律したチームに近づけられることを期待した。

（1）モチベーションの低下

担当プロジェクトの開発手法はウォーターフォール型のため、スプリント単位に動くものができるわけではない。そのため、開発者が開発の完了が見えず、日々タスクだけを消化することになり、開発者が主体的に作業できていない。

（2）情報の共有不足

スクラムは情報を共有することにより他者が気づき改善が行われることが重要であるが、改善が行われる以前に、情報の共有不足により開発の手戻り等が発生した。また、年配者がいることで若手メンバが発言しづらい雰囲気を払拭することができていない。

3.改善前の状態をもたらした原因（因果関係）

「モチベーションの低下」「情報の共有不足」の問題をウォーターフォール型開発時の問題点と比較をおこなった。比較の結果、これらの問題はスクラムの適用により発生した問題ではなく、ウォーターフォール型開発から発生していた問題であることを理解した。スクラムの適用だけでは期待していた効果を得ることができていなかった。

リーダが何かを変えないといけないと思い参加した研修により、原因は特定された。この2つの原因を引き起こしていたのは、「やり方」ではなく「考え方」であった。リーダ及びチームの考えがウォーターフォール型時代のままであることから、「やり方」を変えただけでは期待した効果を得られなかったと判断した。

4.計画した変更内容

外部からの意見を取り入れ、まずはマネージャー及びリーダが「何故しないのか」から「どうしたらやってくれるのか」に考え方を変えた。マネージャーとリーダで2つの問題について「考え方」を変えて対策を検討し、以下の取り組みを実施した。

（1）モチベーションの低下

- ・チームスクラムの目的を伝えるための「共有会」の実施。
- ・チームのやらされ感を排除するための「タスクの扱い」の変更。

（2）情報の共有不足

- ・共有力を鍛えるための「ふりかえり会」の改善。

5. 変更の実現方法

マネージャ及びリーダーの「考え方」を変え、チームと共に実施した活動を以下に示す。

(1) モチベーションの低下

チームはウォーターフォール型開発を経験してきたことから、スクラムに対する目的意識がなかった。まず、スクラムの目的をメンバと共有するための共有会を開催した。共有会では、ワールドカフェ形式で「なぜスクラムをやっているのか」をテーマにメンバの意見を引き出した。この中で、一番多く出た意見が「言われているからやっている」であった。やはり、メンバは上から言われているから実施している思いが強く、メンバの主体性はなくスクラムは運営されていた。この共有会の後に、マネージャが野中郁次郎氏の理論やスクラム導入の目的を情熱的に説明した。これにより、メンバのスクラムへの目的意識が強く芽生えたものとする。

次に、チームが主体的に開発作業を行えていなかった課題について、導入当初は工程の期間が長いためウォーターフォール型開発と同様に、リーダーがタスクを見積りメンバに依頼する形をとっていた。しかし、スクラム導入から半年後あたりに、メンバから「ゴールが見えない」との意見が出始めた。これを改善するために、まず、スプリント単位のバーンダウンチャートと工程単位のバーンダウンチャートを作成し、工程単位のバーンダウンチャートにより、メンバが開発案件の全体進捗を把握できるようにした。しかし、工程単位のバーンダウンチャートは期間が長い（1～2か月）ため、計画変更により実績と予定に大きく差違が見られるようになり、メンバから何のためのバーンダウンチャートであるかわからないとの意見が出始めたため、工程単位のバーンダウンチャートは廃止した。そこで、リーダーがメンバに対して全体を把握させるのではなく、チームが全体を把握できるよう、チームが全工程のタスク見積りを実施する形とした。これを実現するために、これまでのノウハウをもとにWBSを再作成し、メンバが分かり易いようにインプット、アウトプットを明確にした。このWBSをチームへ提供し、メンバがタスクを洗い出すことで、チームが開発案件の全体像を把握できるようにした。なお、リーダーは作業の漏れ、スケジュール遅延の有無等を確認する支援作業に徹した。

(2) 情報の共有不足

チームの共有力を鍛えるために、ふりかえり会に発言力強化を目的として追加した。まず、スクラムの全チームで行っていたふりかえり会をチーム単位でふりかえるように変更した。これにより、人数が多い場合に発言しづらい雰囲気を崩した。次に、チーム単位で出た意見を全体チームに共有するために、代表者が全チームに対して、発言する形をとった。しかし、全チームでの意見交換の際に、発言しない若手メンバがいたため、スクラムチームメンバをシャッフルした形で、全体ふりかえりを実施する形に変更した。これにより、代表者ではなく、それぞれのメンバが自分のチームの状況説明、状況への問いに対して発言できるようにした。改善後のふりかえり会の体制を「図 1 改善後のふりかえり体制イメージ」に示す。

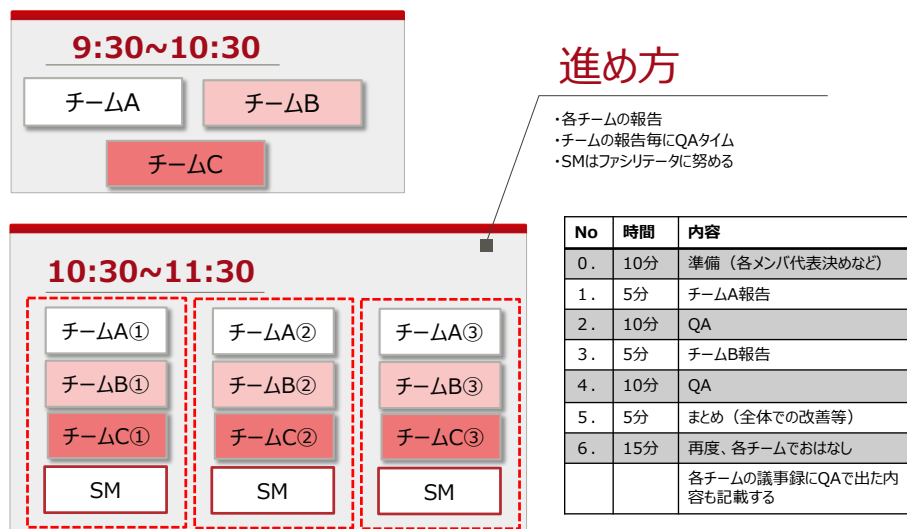


図 1 改善後のふりかえり体制イメージ

6. 変更後の状態や改善効果

変更の実現により、以下の効果を得ることができた。

(1) モチベーションの低下

共有会により、リーダはメンバのスクラムへの想いを知ることができ、何を伝えて推進すべきかが明確となった。メンバは、リーダ及び他メンバの想いを知ること、自律したチームを創りたい想いを共有できたと考えている。スクラムだからコミュニケーションを増やすのではなく、コミュニケーションをとる仕組みとしてスクラムが存在することを改めて考えさせられた。改善することの楽しさを知ること、チーム全体に係る改善がメンバから挙がるようになった。主な改善内容を以下に示す。

図 1 主な改善内容

No	改善	改善内容	効果
1	ニコニコカレンダー	Redmine のニコニコカレンダーを導入	日々の想いを共有できる。話題が作れる。
2	改善フォーラムの追加	改善依頼をできるフォーラムの作成	リーダが欲しいツール作成を要望できる等。
3	Selenium	画面系テストの自動化	結合テスト中にリファクタリングが可能となる。
4	週刊アジャイル	新人メンバが自主的にアジャイルに関する内容を展開	アジャイルに関する情報共有が可能となる。

また、スクラム導入時は、ふりかえり会が形骸化され、半年に 3 回の開催に留まった時期もあったが、現在では、毎週実施しており実施するのが当たり前となっている。これは、リーダが不在でも、メンバが自主的に行動できているため習慣化されたものとする。

メンバへの WBS 展開により、スクラム導入当初にメンバからでいた「ゴールが見えない」との意見もなくなった。また、WBS への項目追加等、メンバから改善意見が出るようになり、想定外の効果も得ている。これは、今までのやらされ感が排除され主体的に作業できている証拠であると思っている。

(2) 情報の共有不足

ふりかえり会の改善により、若手メンバが意見できる環境となっている。これまで、若手メンバがリーダーに意見することは少なかったが、改善後、リーダーのもとに意見を伝えてくれることが多くなっている。また、スクラム導入当初、チームメンバが集まるのは、朝会、夕会、チーム内レビューに限られていたが、改善後は、作業時間内に、ボードの前に集まる姿や、チーム打合せを頻繁に開催する姿が多くみられるようになった。

7.改善活動の妥当性確認

担当チームのスクラム導入目的は、自律したチーム創りである。改善効果で述べているとおり、変更の実現によりリーダーからの支配・制約などを受けずにチームが行動している。これは、チームが自律し始めている証拠だと思っている。自律したことを数値で図ることは難しいが、今後は、KTP の状態や、バーンダウンチャート、ベロシティ等から改善の指標を作成し、評価を行いたい。また、改善前の状態は、導入初期だけでなく、メンバの変更や、慣れによっても引き起こされるものとする。いかに、改善できる環境を継続できる仕組みを構築できるかも今後の課題である。

3B1 「品質保証部門におけるワーキンググループ活動の活性化を目的とした取り組み」 山本美由紀(日立製作所)

<タイトル>

品質保証部門におけるワーキンググループ活動の活性化を目的とした取り組み

<サブタイトル>

なし

<発表者>

氏名(ふりがな)：山本美由紀(やまもと みゆき)

所属：日立製作所

<共同執筆者>

氏名(ふりがな)：河野哲也(こうの てつや)

所属：日立製作所

<要旨>

多様な部署や製品に共通するプロセス・プロダクト上の課題に対してワーキンググループのようなチームを編成し、課題解決を図るアプローチがある。我々の部署でも多様な製品の品質保証業務においてテストの観点を充実させるための課題に対しワーキンググループの形態をとり課題解決を進めている。本発表ではワーキンググループを円滑に進めるための改善事例を報告する。まず改善前のワーキンググループの全体像を示し課題を整理する。そしてワーキンググループの運営体制の改善策を提案し、最後に改善策適用後の効果や考察を述べる。

<キーワード>

改善活動, テスト観点, ファシリテーション, 不具合分析 人財育成

<想定する聴衆>

SEPG 初心者, 品質保証の方

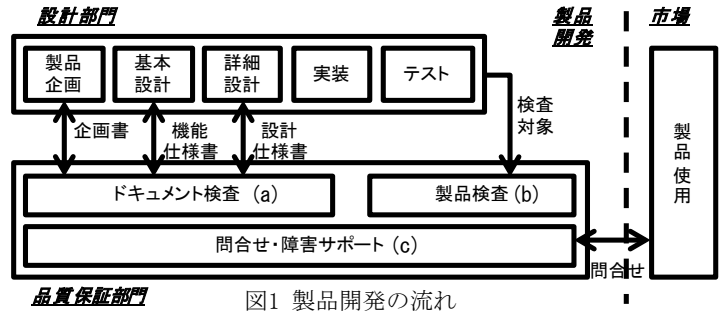
<状況>

- 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()

1. 背景

まず我々品質保証部門の役割を整理するために製品開発全体を図 1 に示す。品質保証部門の役割を大別すると、(a)設計部門の種々のドキュメントをレビューする、(b)製品検査により製品品質を保証する (c) お客様からの問合せや障害に対応するとなる。

筆者は、品質保証部門の支援的な役割の部署に所属し、その役割は以下の 4 つに代弁される。



- ①品質の見える化（品質の状況が見える化し、定期的に関係部署へ情報を提供する）
- ②インフラ整備（見える化を実現するためのインフラを整備する）
- ③品質向上するための施策の支援（検査業務の課題に対しての改善活動の支援と改善活動を通じた技術者の教育支援）
- ④教育の運営（新人導入教育/改善事例報告会/講演会など）

筆者の主な業務は③であり、本発表では③の取組みについて報告する。

ここで品質向上施策とは、各製品や製品検査チームをまたがった様々な共通課題に対して、部内全体から選任者を選出し、小規模ワーキング（WG）を結成し、部内の業務改善を進める活動であり、具体的には次のような形式で進めている。各製品検査チームに潜在する課題の共通性やその根本的な原因を分析し、それらの課題解決のために改善策およびその運用法を立案し、それらを各選任者の製品検査チームで試行・評価し、さらにそれらの結果をフィードバックするというサイクルを回し、品質向上・業務改善を行っている。

品質向上施策には複数 WG があり、筆者はそれらの WG が円滑に進むように運営支援を行っている。複数の WG の 1 つにテスト観点知識ベース WG があり、本稿ではこの WG に焦点を当てて、以降議論する。

ここでテスト観点知識ベース[1]（以降 TKB と呼ぶ）とは、各市場不具合事例に対して、それらを検出するためにはどのようなテスト観点が必要だったかを分析・抽出し、各製品に依存せず横断的に活用することを目的とし、それらを整理・蓄積した知識ベースのことであり、テスト観点知識ベース WG（以降、TKB-WG と呼ぶ）では、TKB の保守や活用を推進しており、今回報告する TKB-WG の活動は TKB に対してテスト観点を拡充していくことを主たる目的としている（TKB そのものの考え方や有効性などの詳細については本稿では論述しない。[1]を参照）。

この TKB-WG を運用するにあたり、「TKB へのテスト観pointsの登録が進まない」という課題があった。本発表では、この課題を解決するための WG の体制や運営に関しての改善の取組みを報告する。

[1]光永 洋、田中 浩和、「故障事例によるテスト観点知識ベース構築とテスト設計への適用」、ソフトウェア品質シンポジウム 2012

2. 改善前の状態

まず、改善前の TKB-WG の全体像を図 2 に示し、TKB-WG の概要について述べる。

TKB-WG は、メンバとアドバイザーという 2 つの役割によって結成され、筆者は WG が円滑に運営できるように支援を実施している。また、TKB-WG の活動は、毎週 2 時間程度を目安に開催している。メンバは各製品検査チームにおいて実際に検査を実施している技術者から選任され、合計 10 名程度である。そして、メンバは WG

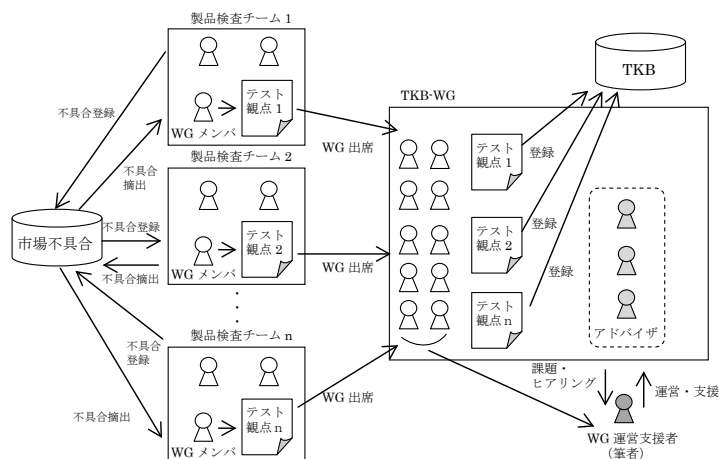


図 2 TKB-WG 全体図（改善前）

活動の一環として各検査担当製品の市場不具合を分析しテスト観点を抽出する。そして、WG には抽出したテスト観点を持参し出席する。WG では、メンバの人財育成もかねて、メンバが主体となり、各自持ち寄ったテスト観点について議論し、テスト観点のブラッシュアップを行う。アドバイザーは各製品検査チームのリーダーから選任し合計 3 名であり、WG には出席するものの議論を主導せずあくまでもアドバイスをする立場として参加し、またブラッシュアップ後のテスト観点については TKB 登録可否を判断する。そのように登録可否を判断され、合格となったテスト観点が TKB に登録される。また、WG 運用・支援者（筆者）は WG メンバとコミュニケーションを図り、WG メンバの課題や、漠然とかかえている問題をヒアリングした上で整理し、WG の運営・支援を行う。

以上の活動において、次のような課題があった。WG において、メンバの主体性に任せテスト観点に関して議論・ブラッシュアップを行っていたが、議論そのものが上手く発展しないためテスト観点のブラッシュアップが行えず、その登録が進まない課題が顕著に表れるようになった。加えて、メンバ・アドバイザーを含めて 13 名が毎週 2 時間を確保することが難しく、欠席者が目立つようになってきたため、WG の一体感の醸成に欠け、それも議論が活発にならない原因の一つであった。

3.改善前の状態をもたらした原因（因果関係）

TKB-WG では、テスト観点の拡充を目的として運営しているが、WG 活動を通じたメンバの人財育成ももう一つの目的としている。そのため、メンバの自主性に期待し、各メンバ自身が議論を進展させ、また議論を整理する能力を身に付けることを狙い、WG の運営を行ってきた。

しかしながら、上記の想定と外れ、WG の議論は活発にはならなかった。WG 活動の様子から、メンバは議論そのものに対してやる気のないわけではなく、アドバイザーが時折議論発展のためのアドバイスをすると、議論が活発になることが見受けられた。一方、WG 活動以外に目を向けてみると、各メンバは担当製品検査チーム内では活発に議論しているもののチームの外で議論する機会、例えば改善事例報告会の質疑などでは議論が活発になることが少なく、我々品質保証部門の技術者は議論することに慣れておらず、また組織的に議論する素地が少ないのではないかと考えた。

また、欠席者の多くは、担当製品の検査期間と重なることや突発的なお客様から問い合わせや障害対応が発生することが原因であることがわかり、これらの原因は WG ではコントロールできないことがわかった。

4.計画した変更内容

変更後の TKB-WG の全体像を図 3 に示す。以降、図 2 からの大きな変更点について述べる。

まず、TKB-WG を 3~4 名のメンバで編成するサブ WG に分け、メンバのスケジューリングを容易にした。

次に、3. で述べたように、WG において議論を活発化させるためには、ファシリテータが必要であると考えた。そのため、WG のアドバイザーがファシリテータの役割を担い、WG そのものをサブ WG に分け、各サブ WG に 1 名のアドバイザーをアサインするようにした。加えて、TKB へのテスト観点の登録をサブ WG の成果として捉え、チームの一体感の醸成を図るようにした。

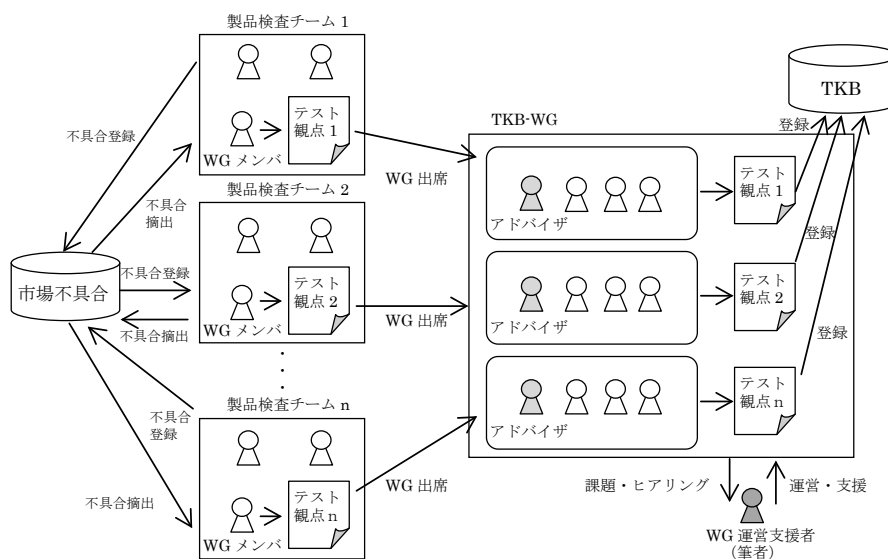


図 3 TKB-WG 全体図 (改善後)

5. 変更の実現方法

以下、変更後のTKB-WGにおける工夫点について述べる。

まず、テスト観点は各製品に依存せず横断的に活用することを目的としているため、製品横断でサブWGのメンバを編成し、担当製品以外の事例やテストの工夫などの知見を得る機会を設けた。ただし、サブWGに分けたことで、サブWG間で重複した市場不具合事例を分析する可能性があったため、運営・支援側で事前に分析対象の市場不具合事例を選定するようにした。

次に、サブWGの開催は、事前にスケジュールリングするものの、種々の事情で出席できないメンバが出てきた場合は、その都度、再度スケジュールリングを行い、極力サブWGの全メンバが出席できるように工夫をした。

加えて、運営支援側の工夫として、各自および各サブWGのテスト観定の毎月の登録数を集計し、見える化することで、意識の向上を図った。また、TKB-WGに一定期間所属し、ある一定数のテスト観点を登録したメンバには、テスト観点マイスタというテスト観点に関しての専門知識を保有する技術者として認定するようにした。

6. 変更後の状態や改善効果

ファシリテータとして確実にアドバイザーがサブWGに出席するようにし、かつ各メンバもほぼ全員出席できるようになったことで、テスト観定のブラッシュアップからTKBへの登録までの一連の運用が定常的に実施され、テスト観定の拡充が大きく進むようになった。

また、WGメンバの約10名がテスト観点マイスタとして認定され、WG活性化に貢献するようになった。

7. 改善活動の妥当性確認

表1に変更前と変更後のテスト観定の登録数を示し図4に変更前と変更後のテスト観定の登録数のグラフを示す。

表1 改善前と改善後のテスト観定の登録件数

	期間	テスト観点登録件数	WGメンバ数	1人あたりの1か月のテスト観点登録件数
改善前	13.6ヶ月 (2013.10.01～2014.11.20)	451 観点	10 人	3.2 観点/人月
改善後	4ヶ月 (2014.11.27～2015.03.30)	160 観点	10 人	4.0 観点/人月

表1・図4より、変更前より変更後のほうが1人あたりの1か月テスト観点登録件数が増加したことがわかる。なお、大きな増加がみられないのは、改善後の期間が改善前と比較して短いため、サブWGの運営に慣れていないことが考えられる。今後、改善後の体制で運用が進めば、大きな効果が期待できる。

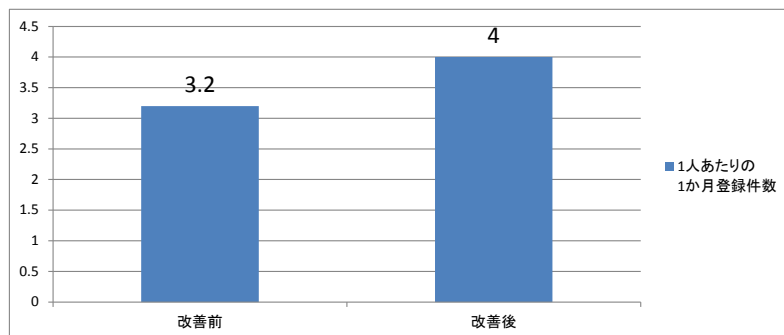


図4 1人あたりの1か月テスト観点登録件数推移

3B2 「SPI 経験を活かした SQA 活動実践」 植村貴美(パナソニックヘルスケア)

<タイトル>

SPI 経験を活かした SQA 活動実践

<サブタイトル>

－SQA 1 年生の取組み－

<発表者>

氏名（ふりがな）： 植村貴美（うえむらたかみ）

所属：パナソニックヘルスケア株式会社 診断薬事業部 品質管理部 機器品質保証課

<共同執筆者>

無し

氏名（ふりがな）：

所属：

<要旨>

医療機器開発メーカーの全社 SPI 行政担当業務(技術職能)から事業部 SQA(担当品質職能)へ職種変換し、事業部内での SQA 活動強化に取り組んだ SQA 一年目の活動事例。
SQA 手順書の作成、SEPG との連携強化、他社 SQA との交流等、これまでの SQA 活動の延長線上の活動に加え、開発現場でのピアレビュー参加や、進捗会での課題共有など、自身の SPI 経験を活かした活動についても紹介する。

<キーワード>

SQA、SPI、ソフトウェアプロセス、業務の見える化、SEPG

<想定する聴衆>

品質保証の方、SPI の方、SEPG

<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

弊社は、血糖測定器や薬用保冷庫、電子カルテシステムなど、ヘルスケアに関する機器やシステムの開発製造販売を行っており、所属する診断薬事業部では、血糖測定器やセンサーの開発製造を行っている。医療機器開発においては、法律、規格要求、品質システム要求など、制約事項への対応が必須条件となっている。製品自体だけでなく、そこに組み込まれるソフトウェアの開発に対しても、開発プロセスを遵守していることを示す必要があり、ソフトウェア開発基準書を整備し基準に基づいた開発を行っている。開発プロセスに則った開発を行う風土がある一方、現場では、開発効率を上げる必要もあり、開発プロセスが重たいと感じる一面もある。

2.改善前の状態

開発現場は、プロセス(仕組み)はあり、実施もされている(CMMI L2 相当)状態。技術部門にて、S/W技術者が主体となり、自主的に SEPG 活動が推進されており、外部監査等での指摘事項の共有などは徹底されているが、具体的な PJ 内の課題共有までは行われていない。一方、品質部門においては、SQA 担当者は 1 名アサインされており各開発ステップ移行時の監査活動は実施されているが、QA 業務との兼任のため工数は不足気味であり、PJ 内の改善活動状況までは把握できていない。SQA 担当者も SEPG 活動メンバーとなっているが、参加できない場合も多々あった。

3.改善前の状態をもたらした原因（因果関係）

外部機関の監査指摘については、組織として是正が求められるため、SEPG 活動を通して対応がされてきたが、SQA が実施する内部のステップ移行監査では、PJ 毎に対応をとるため、その指摘内容に対する共有が不十分である。また、医療機器開発においては開発サイクルが比較的長いものが多く、新規開発の場合は企画から出荷まで数年にわたる場合がある。そのため、PJ 間でプロセスを回す経験値に差が生じ、頻繁にプロセスを回す PJの方が効率的にプロセス運用されるという温度差に繋がっている。また、基準書は各 PJ 共通であるが、エンジニアリングに関するプロセスは PJ 毎に異なることも PJ 間の温度差に影響を及ぼしている。一方、SQA 担当者は、アサインされているものの QA 業務との兼務であり、慢性的な工数が不足気味のため、開発プロセスが実施されていることの確認はできているが、効率的にプロセスが運用されているかまで見届けることができていない。

4.計画した変更内容

(1)SQA 手順書(品質保証プロセス手順)の作成

【目的(ねらい)】

- ・SQA 業務の平準化とコミットメント：SQA の増員に伴い、SQA 業務の見える化を行い、作業の平準化を図るとともに、品質管理部門にて SQA 業務工数に対するコミットメントを得る。
- ・SQA に対する要望の収集と対応：手順書作成の過程を利用して、SQA 業務についての議論の場を設け、SQA に対する要望を収集して対応策を検討し SQA 手順書に反映する。

(2)PJ 開発現場への関与

【目的(ねらい)】

- ・PJ 間のプロセス運用スキル差の改善：経験の浅い PJ においても開発プロセスを効率的に運用できるようプロセス運用スキルのボトムアップを図る。

・PJ の状況把握：開発ステップ移行時の監査では、PJ 課題まで把握することは難しい。また、タイムリーに課題を発見することも困難である。PJ 現場の負担が少ない方法で課題を収集し、必要なアラームをあげることで、課題の早期解決を図る。

5. 変更の実現方法

(1) SQA 手順書(品質保証プロセス手順書)の作成

- ① 現 SQA 担当者から現状の SQA 業務をヒアリングし、現在の SQA 業務を描き表すことで現在の SQA 業務の実態を把握する。
- ② SQA 業務の改善項目を抽出するため、①と並行して SQA のあるべき姿を SQA 業務指針として明らかにする。
- ③ ①と②を比較し、現在の SQA 業務における改善点を SQA 手順書に盛り込む。
- ④ SEPG 活動にて SQA が中心となって、SQA 手順書策作成 WG を立ち上げ、手順書作成の進捗を報告するとともに、SQA 手順書のレビューを実施。
- ⑤作成した手順書を事業部の QMS に登録し運用。

【主な工夫点】

・議論の場作り: 上記④

SEPG 内に SQA 業務を議論できる場を設けた(WG の立ち上げ)。

レビューに品質管理部門の上級管理層(SQA 担当者の直属上司)も巻き込み、技術部門と品管部門で情報を共有。

レビューの中で技術側の SQA に対する要望の引き出し。

・SQA 手順の粒度：上記①

SQA 手順の記載粒度を大きめにし、PJ へ柔軟に対応。(PJ 毎に開発プロセス手順が異なり、プロセス設計を行うため)

手順の粒度が大きくなっても、作業の質を確保するため、開発計画書と品質計画書を対比させることで、PJ で計画した内容が適切に実施されているか確認することを周知。

(2) PJ 開発現場への関与

- ① PJ の現在の開発フェーズ、過去のプロセス運用実績、開発体制を調査。
- ② ①と並行して、S/W 技術者のマネージメント層に、SQA に対する要望や目指している姿をヒアリング。
- ③ ①と②を踏まえて、PJ 毎に支援スコープと支援内容を決定。
- ④ PJ の開発フェーズの切り替わり時期から段階的に PJ 現場への SQA の関与を開始。

【主な工夫点】

・PJ の実態把握：上記①

開発計画書、PJ のふりかえり資料等のドキュメントからの情報収集にとどまらず、現場の開発メンバーとコミュニケーションを取る。

S/W 開発の枠組みにとらわれず、H/W 開発体制も含め製品開発全体にも目を向ける。

・プロセス運用スキルによる支援内容の判断：上記③

プロセス運用スキルの高い PJ のノウハウを水平展開し、全体のボトムアップにつなげる。

水平展開する内容は、使用する帳票や記載方法など具体的なものにする。(効果の実感が得や

すい)

進捗会にて PJ 課題を収集する際、PJ のレベルに合わせてスコープを変える。(S/W 開発 or 製品開発)

・プロセス運用スキルの判断材料：上記③

プロセス運用スキルの判断基準として、PJ メンバー構成、S/W リーダーの経験値を考慮し、CMMI の PA の観点も役立てる。

S/W 開発プロセスが効率的に回っている PJ では、システム(製品)開発にスコープを広げて課題収集。

6. 変更後の状態や改善効果

SQA 手順書を事業部 QMS に登録し、2015 年度 # 4 四半期より運用開始、現在は SEPG 委員会にて改善情報を収集中。SQA 業務指針と SQA 手順書を作成したことで、あるべき姿と現状の見える化ができ、今後の改善活動に役立てていける。

開発完了 PJ が少ないため PJ 毎のふりかえりはできていないが、SEPG の 2014 年度の活動ふりかえりにおいて、いずれの PJ からでも 2014 年度は SQA の PJ 関与度が向上、監査時の指摘内容が良くなったとのコメントをもらっている。SQA の PJ の進捗会への参加に対してもどの PJ からでも高い評価を得ているので、今後も PJ のレベルに合わせたスコープでの参加を継続したい。

7. 改善活動の妥当性確認

限られた SQA の工数を効率的に現場 PJ の支援に回すため、PJ のプロセス運用スキルに応じたアプローチを試みたため、改善活動のボトムアップに繋がった。今後は、プロセス運用スキルの高い PJ に対して、改善提案するために、S/W の枠組みを超えた活動が課題となる。

3B3 「なぜなぜ分析におけるファシリテータの課題と改善策の紹介」 林真也(住友電工情報システム)

<タイトル>

なぜなぜ分析におけるファシリテータの課題と改善策の紹介

<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 第一システムグループ

<共同執筆者>

氏名（ふりがな）：

所属：

<要旨>

CAR（原因分析と解決）プロセスを効果的に実施する上で阻害要因になっている、根本原因の分析がなぜうまくできないかを分析したところ、38件の課題を抽出した。これらの課題の内13件は、分析の会議体におけるファシリテータ起因の課題であった。ファシリテータの課題を解決するため、実務で発生した不具合等を実際に分析して、スキルアップを図る方法をとった。その実践方法と効果について報告する。

<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

なぜなぜツリー 真因 根本原因

<想定する聴衆>

改善活動に関わる管理職

<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。それに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とする。

2.改善前の状態

各プロジェクトの成功事例・失敗事例は弊社内で定期開催している品質大会、随時報告しているトラブル報告、という手段により公開され、他プロジェクトメンバーにも共有される仕組みを有している。それら報告内容にはよく分析しているものもあれば、不十分な分析で終わっているものもあり、分析の実施者や事象の複雑さにより、分析的確性に差異が存在する。不十分な分析で終わっていると、実現可能で有効な対策が打てず、同様の失敗を再発する要因にもなっている。

しかしながら的確に事象を分析できる人材が不足しており、現時点の能力ある分析実施者だけでは、プロセス改善による組織の目標達成が確実視できない。

プロセス改善を強力に推し進めるには、的確な原因分析ができる人材を増やすことが急務である。根本原因が見つければ、効果のある対策を立てやすくなり、組織のプロセス改善に寄与しやすくなるであろうと期待する。また、人材育成は一時的なものではなく繰り返し計画的に実行する。これにより裾野を大きく広げて、あらゆるプロジェクトの問題点を解決できるようになることも期待する。

3.改善前の状態をもたらした原因（因果関係）

何が事象の分析がうまくできない原因になっているのかを知るため、9名の有識者でなぜなぜ分析をした。議論をするにあたり、「なぜなぜがうまくできない状態」「よくないなぜなぜのO/Pの状態」とはどういうものかを先に描いた。それを視覚化したものが図1の①から④の4状態である。それぞれ下記の①から④がその説明である。図中に無い⑤は、①から④の議論中に発生した共通の問題を分けて足したものである。

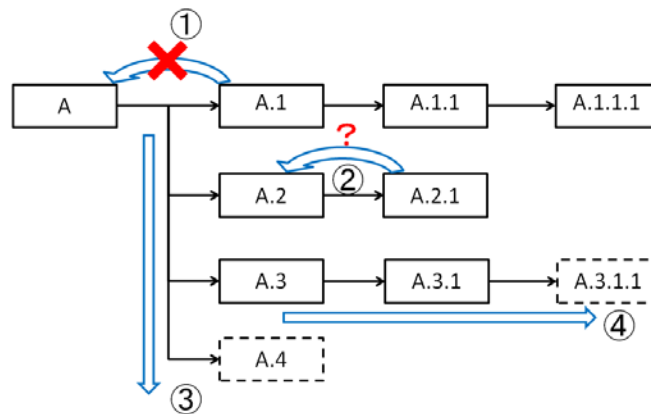


図1. よくないなぜなぜツリーの要素

- ①：原因と結果の連鎖において因果関係が成立していない
- ②：因果関係は成立しているが、確実性を疑う原因を記述している
- ③：上位の事象の原因がすべて上げられていない
- ④：原因の掘り下げで根本まで到達していない

⑤：分析時間が足りていない

分析のための会議は10回（1回当たり2時間弱）におよび、38件の問題点を抽出した。
38件の問題点は表1の通りである。

表1. なぜなぜ分析で発見した38件の問題点

(1)発言が適切な表現になっていないときに聞き返していない
(2)発言内容を適切な表現に言い換えて確認していない
(3)原因を1つずつ発言させていない
(4)下から上への因果関係の確認が出来ていない
(5)事実・推定を明示する手順や様式になっていない
(6)計画していない
(7)発動に関する指針がない
(8)思い出す材料(例:行動記録やいろいろなログ等)がない
(9)思考プロセス(帰納法、演繹法)が身につけていない
(10)分析者の構成にプログラマが少ない
(11)ファシリテートの進め方で最初の質問で作り込みを聞けていない
(12)ファシリテートの仕方を教えていない
(13)ファシリテータがコントロールできていない
(14)あらゆる要因・観点をあらかじめ示していない
(15)有識者がメンバーに含まれていない
(16)根本原因の定義を具体的に教えていない(例:プロセス、教育)
(17)(実用的な)根本原因そのものを定義していない
(18)掘り下げを止める基準がない
(19)掘り下げを続ける基準がない
(20)ファシリテーターがその場で是正していない
(21)現物を見て確認していない
(22)誰が参加すべきかの選定基準がない
(23)誰が参加すべきかの選定基準がない
(24)なぜなぜ会議の出席者の役割が定義されていない
(25)トレーニングがされていない
(26)ベースプロダクト視点を示していない(例:から)
(27)原因のレベル毎(対策の波及範囲の広さ)に効果・コスト・リスクのバランスで取捨するプロセスがない(例:から)
(28)組織全体でバックアップする仕組みがない
(29)分析する事象を選択していない
(30)冒頭に経緯説明を入れていない
(31)事前に資料配布していない
(32)ファシリテータがコントロールできていない
(33)経緯(関連しそうな事実の時系列的な列挙)を文書化していない
(34)一次原因を分析して文書化していない
(35)現物(ソースコード、仕様書)があるのに口頭で説明をする
(36)会議の目的を十分に説明していない
(37)ファシリテータがコントロールできていない
(38)「なぜ？」という質問自体が糾弾になっている(ネガとポジの使い分け)

4.計画した変更内容

問題点38件の多くが「分析プロセス定義」(20件)、「ファシリテート能力」(13件)の2つに帰することを見つけた。

構想から対策の効果測定までを考えたとき、前者の分析プロセス定義は、有識者による提案から組織への浸透まで半年ぐらいかかる見込みである。後者のファシリテート能力は、有識者から現場（なぜなぜ分析会議の実施者）への説明と変化点の収集で、3ヶ月ほどで効果が測れそうである。さらに、改変後のなぜなぜ分析会議からフィードバックを受け、前者の分析プロセス定義に反映できるものが存在するはずである。効果の早い測定と、有益なフィードバックが得られるであろう期待から、先にファシリテート能力の問題を解決することにした。

今回は「ファシリテート能力」向上を課題として取り上げ、対策は問題点13点それぞれの裏返しとした。この新しい対策を説明する対象者を10人程度に絞り込み、教育し、構想から3ヶ月以内に「ファシリテータスキルが身に付いたか？」を評価する。またこの対策が「教育として効果が見込めるものか？」についても評価する。

5. 変更の実現方法

改善の方向性が定まったところで、次は具体的な教育プランをつめなければならない。教育は講習会よりも、本物のなぜなぜ分析の現場を活用させることにした。現実の問題のほうが具体的であり分かりやすく、分析により根本原因が特定できれば、実業務の役にも立つという効果も期待できるからである。

数回なぜなぜ分析の場を設けて、高い分析能力を漸進的に身につけてもらうようにした。分析会議毎にアドバイザーからファシリテータへ現在どの能力レベルにあるのか、および会議内での気づきを伝え、次回により高いレベルを目指すように意識させる工夫を考えた。

具体的には、有識者のチームで以下のような教育プランを考え、実施している。

受講者：	各部署からの選出 総勢 9 名
支援部隊：	有識者のチームがアドバイザーとして分析会議に入る。
分析テーマ：	開発／保守のプロジェクトで発生した現実の問題を対象にする。
期間・頻度：	2ヶ月間で分析会議を 6 回。1 回あたりの会議時間はファシリテータに委任。
採点：	以下の評価ができる記入シートを作成。会議毎に収集する。(図 2 参照) ・定量評価 15 項目 (1 点～4 点、および評価対象外) ・定性評価 KPT 方式による気づき
教育の評価：	採点結果を使い以下の 2 点の問いかけに絞って判断する。 ・根本原因を見つけられたか？ ・意図したファシリテータスキルが身に付いたか？

ファシリテーター:
アドバイザー:

【スキル評価】

ファシリテータースキルについて判断をお願いします。各項目に評点(※)を記入してください。

※評点: 1でできていない 2あまりできていない 3ほぼできている 4できている -評価対象外

分類	評価項目	評点		
		本人	アドバイザー	
準備	1 会議の冒頭または事前に事象の経緯を参加者に理解してもらっている			
	2 参加者は適切であった			
実践	因果関係の正しさ	3 発言が漢語でなければ聞き返す、あるいは漢語に言い換えて「言ったことは『〇〇が××を△△した』ということですか?」と確認する。		
		4 なぜなぜツリーに1文書いたら、上位文と繋げて読み上げて因果関係の正しさを確認する。「〇〇(書いた文)だから××(上位にある文)」でいいですか?		
		5 原因文が具体的ではない場合(例:レビュー不足でした)は具体的にまで聞き返す。		
		6 現物(プログラム、仕様書、手順書)を見て、具体的な行動を聞いて、発言の真偽、具体性、表現適切性を確認する。		
		7 原因は同位方向(深く入らず同じレベル)に先に出していく。(上位文を指して)「これらの原因で全てですか?」「〇〇、××、・・・(列挙された原因)が解消するとこの事象(上位文)は発生しませんか?」などと尋ねる。		
		8 原因(事象)は1つずつ発言させる。複数の原因や、原因の連鎖を同時に発言した場合は1つずつに留めてもらう。		
		9 分析箇所(その時原因を尋ねている文)以外の原因の発言は制止し後で発言してもらう。		
	因果関係の発点の委縮	10 自己防衛的な言い訳(仕方ない)や逆に自虐的・自省的な謝罪(すいません、私が悪い、私の知識不足)のような発言は制止し、それ以外の観点の原因を尋ねる。		
		11 欠陥分析の場合は「見逃し」原因のみになりがちなので、「作り込み」原因を先に聞き出し(質問し)分析する。		
		12 何を見て作業したか?(入力成果物の観点)、ツールやFWに制約があったか?、など別の観点の気づきを与える。		
	成果物	13 多くの発言、多くの観点を引き出している(ツリー中の観点の多さや、同位方向の観点の多さ)		
		14 因果関係が成立している		
		15 根本原因を特定している		
		得点	0	0

【KPT】

Keep	
Problem	
Try	

図2. 実践教育で使う採点シート(フィードバックシート)

6. 変更後の状態や改善効果

受講者 9 名の採点を表に転記して、相加平均を算出した（表 2）。計画している回数に到達していない受講者がいたが、受講者の半数が 4 回目を終えていた。全体像を知る上で相加平均の値を利用して評価できると判断した。平均値をグラフ化したものが図 3 である。回数別カテゴリ別の成長具合と長所・短所が分かるようにレーダーチャートで示している。

表 2. 受講者の採点

ファミリデータ	採点カテゴリ	第1回目	第2回目	第3回目	第4回目	第5回目	第6回目	個人平均
A	準備	3	3	4	4	4	3.5	3.58
	因果関係の正しさ	1.6	2.8	3	2.6	3	3.2	2.70
	会議のコントロール	1	1	2	2	3	3	2.00
	分析観点の気づき	2.5	3	3	2	3.5	2.5	2.75
	活性化	2	2	2	2	3	4	2.50
	成果物	3	2	2	3	4	3	2.83
B	準備	3.5						3.50
	因果関係の正しさ	1.4						1.40
	会議のコントロール	2						2.00
全体平均	準備	3.33	3.38	3.06	3.75	4.00	3.50	3.30
	因果関係の正しさ	1.84	2.50	2.38	2.50	3.00	3.20	2.22
	会議のコントロール	2.00	2.31	2.52	2.68	3.00	3.00	2.39
	分析観点の気づき	2.56	2.50	2.88	2.88	3.50	2.50	2.58
	活性化	2.11	2.13	2.25	2.25	3.00	4.00	2.34
	成果物	2.67	2.88	2.63	3.00	4.00	3.00	2.77
	活性化	3	4	3				3.33
	成果物	4	4	4				4.00

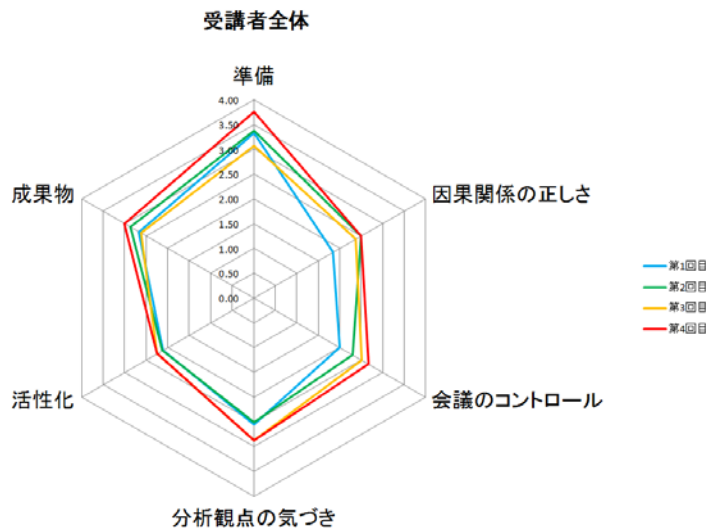


図 3. 回数別カテゴリ別受講者の採点チャート

4 回目の採点結果がすべてのカテゴリに対して最大であり、また回を追うごとに面積が徐々に成長してきていることが分かる。ただ、成長の度合いはカテゴリにより異なり、実践トレーニングで成長しやすいものと成長しにくいものがある。

合格ラインを 3 点としたとき、現時点でこれを満たしているものは「準備」と「成果物」の 2 つである。合格ラインに近いものは「会議のコントロール」と「分析観点の気づき」であり、成長の伸びを見る限り 6 回目完了時には達成の可能性が高い。

一方「因果関係の正しさ」と「活性化」は合格ラインからまだ遠く、特に「活性化」については最初からの伸びが他と比べても非常に小さい。受講者個別で「因果関係の正しさ」について見れば、初回から伸びている

人もいれば、低く止まっている人もいる。トレーニング中にコツをつかむ何かがあったと思われるので、調査が必要である。「活性化」も伸びている人と伸びていない人がいるが、総じて伸びが芳しくない。現在の教育方法では伸ばすことが難しいのかもしれない。今後に分析・調査が必要である。

5項で示した教育の評価に関しては、実践が6回に達していない受講者が多いため、最終判断はまだできていない。

7.改善活動の妥当性確認

ファシリテート能力が当初より向上していることから、今回の教育プランは効果があると判断する。その上で、次の3点の課題がある。

- 1つ目は今回の教育プラン実施で成長が十分でないカテゴリーの原因の追究である。
 - 2つ目は受講者からのフィードバックを教育プランまたは分析プロセスにどう反映させるかである。
 - 3つ目が「分析プロセス定義」の具体的な対策立案と実施である。
- 引き続き有識者で検討・対処して、プロセス改善に努めてゆく。

3C1 「約 2 年半で CMMI ML3 を達成した部門の SPI 活動事例紹介」 小田倉泰浩(東芝デジタルメディアエンジニアリング)

<タイトル>

約 2 年半で CMMI ML3 を達成した部門の SPI 活動事例紹介

<サブタイトル>

<発表者>

氏名 (おだくら やすひろ) : 小田倉 泰浩

所属 : 東芝デジタルメディアエンジニアリング株式会社

<共同執筆者>

氏名 (ふりがな) :

所属 :

<要旨>

『SPI』を全く知らない組織が、約 2 年半という短期間で CMMI ML3 を達成しました。

そして現在も SPI 活動を継続しています。

この短期間で経験した SEPG、SQAG の立ち上げやトレーニングの実施などについて
失敗や苦労を成果効果のグラフを織り交ぜながら説明しつつ、そこから学んだオーナーシップ
の大切さなどを紹介します

<キーワード>

SPI 初心者、CMMI ML3、成果効果、管理層、オーナーシップ

<想定する聴衆>

SEPG 初心者、管理層

<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

当社は東芝製品の開発を生業としているエンジニアリング会社です。主に PC、TV、携帯電話などデジタルプロダクツの開発を担当しておりますが、2008年リーマンショックなどによる事業環境の変化から、当部門としても、新規ビジネスの開拓が急務となりました。

2.改善前の状態

新規顧客を開拓し受注するも、新規プロジェクトではトラブルが発生することが多く、深夜残業・休日出勤など人海戦術でなんとか乗り越えるという状態。現場には『なんとかしたい』との思いがありました。

3.改善前の状態をもたらした原因（因果関係）

当時は個人のスキルに依存しており、組織としてプロセス改善の取り組みが、やや弱かったように思えます。

4.計画した変更内容

このような中、2011年に当事業部のトップが交代しました。

新事業部長より『CMMI ML3を2年で達成せよ』との強烈的なチャレンジがあり、また現場の『なんとかしたい』という強い気持ちとも一致。小中規模が中心の小さな部門ですが、SEPG、SQAGを部門のメンバが兼務し活動する計画を実行立案しました。

はじめに ~ 私たちがSPI活動を始めたきっかけ ~

**CMMI ML3
2年で達成せよ。**

エンベテッドシステムグループ
GM (事業部長)
2011.6 就任

売上高

トラブル

深夜残業

◆小中規模PJが中心 ◆CMMIが更なる負荷にならないかと不安
◆現場は困っている『なんとかしたい』

2011.10 エンベテッドシステム担当
“[ES]”に改名、管理層もほぼ一新、
本気で改善に挑む姿勢

2011.12 [SWP]様
コンサルのもと、SPI定
例会がスタート!

TOSHIBA
Leading Innovation

Copyright2014 Toshiba Digital Media Engineering Corporation

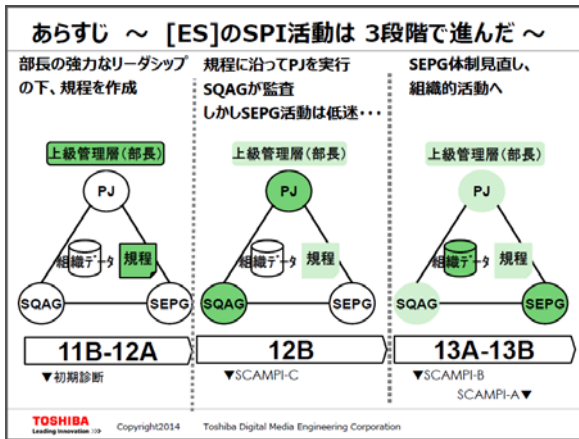
5.変更の実現方法

コンサルティングの支援を頂きながら、2011年12月にSPI活動を本格的に開始。

CMMIなどのトレーニングを受けつつSEPG、SQAGを段階的に立ち上げました。

当初は、トラブルプロジェクトをいくつか抱えており、SPI活動を兼務で進めるのは非常に辛い作業で大変でしたが、部長の強くなりーダシップのもとに推進していきました。

活動開始から1年後には新しい規程を適応したパイロットプロジェクトが試行され、SQAG監査も開始されるまでに成長しました。しかしながら、SEPGは参加率が低下してしまい、活動低迷の中だるみ状態へ・・・



これを打開すべく、SEPG の体制を大幅に見直し『課長+若手』の構成として、心機一転を図りました。権限を持つ課長と、俊敏に動ける若手の組合せは、上手く機能し、約 1 年後の SCAMPI-A にて CMMI ML3 を無事達成することができました。

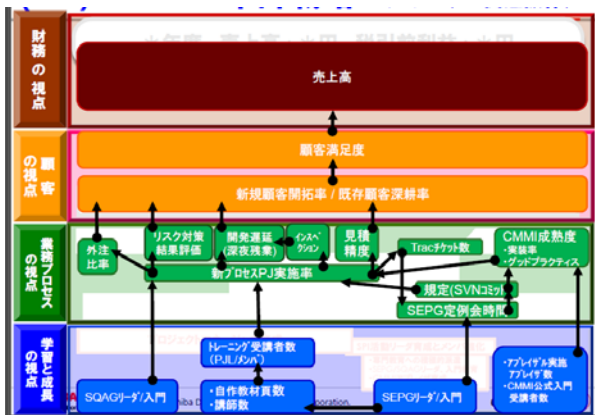
6. 変更後の状態や改善効果

現在、CMMI ML3 達成から 1 年以上が経過しましたが、当部門では SEPG、SQAG とも活発に活動を続けています。またトラブル発生による深夜残業なども、改善前と比較し、減らすことができました。

7. 改善活動の妥当性確認

CMMI ML3 達成後に『SPI 活動の成果効果』を、資料にまとめました。

事業の戦略マップを元に作成し、『学習』『プロセス』『顧客』『財務』のそれぞれの視点について、メトリクスを設定。SPI 活動の開始から CMMI ML3 達成までの経年変化をグラフで示すことで、成果効果を視覚的に表しました。



さいごに、約 2 年半の SPI 活動を通じて、私達が学んだことのうち、特に印象に残った以下 3 点を紹介します。

- 『改善活動ができるようになるのは、管理層の責任』
 - ・初期段階では上級管理層（部長）の強力なリーダーシップが必要
 - ・ある段階で部長は身を引き、組織（SEPG/SQAG/PJ）が主役となるようバトンタッチ
- 『プロセス・オーナー・シップ』改善活動の主役は自部門
 - ・「なんとかしたい」という自部門の気持ちが源泉
 - ・「次の世代」が困らないように『自律した改善活動』の仕組みを作ることが大切
- 小さな組織は強みになる
 - ・PJ メンバが SQAG や SEPG を兼務するのは、最初は大変だが、小回り良くプロセスが浸透し運営は楽になる

3C2 「プロセス改善活動における GQM の評価」 山邊人美(住友電工情報システム)

<タイトル>

プロセス改善活動における GQM の評価

<サブタイトル>

<発表者>

氏名（ふりがな）： 山邊 人美（やまべ ひとみ）

所属：住友電工情報システム株式会社 QCD 改善推進部 プロジェクト管理グループ

<共同執筆者>

氏名（ふりがな）：

所属：

<要旨（150～250 文字）>

組織目標を達成するためにプロセス改善活動を実施しているが、「目標値の妥当性が低い」「施策の妥当性が低い」「試行が目的になっている」「場当たりの評価になっている」等の課題を抱えている。これらを解決する手法として CMU 版 GQM に着目し、その効果を評価した。その結果、上述の課題に対して効果があることが確認できた。また手法の展開に関して教育コスト及び実施コストを評価した結果、特に問題なるコストではなく、組織展開することにした。

<キーワード（5～25 個）>

GQM、プロセス改善活動、目標施策立案、メンタルモデル、要因分析、CSF（重要成功要因）、測定、評価報告書

<想定する聴衆>

改善活動に関わる管理職、SEPG 及び改善活動の実施者

<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

弊社はCMMIレベル5を達成しており2017年の中期計画の目標達成のために改善活動を実施している。各工程毎にWGを設立しプロセス改善活動を進めようとしているが (a) 目標値の妥当性が低い (b) 施策の妥当性が低い (c) 試行が目的になっている (d) 場当たりの評価になっている といった課題を抱えたWGがある。一方、2014年のSPI Japanで発表したように弊社のSS第三システム部では保守業務の評価指標を決めるためGQM手法を活用し、適切な評価指標を設定することができた。これを受けてGQM手法がプロセス改善活動にも効果があるのではないかと考え、その効果を評価することにした。

2. 採用した手法

評価にあたりGQM手法を調査した結果、

(a) メリーランド大学のピクター・バシリ教授によって開発されたオリジナルのGQM手法

(b) カーネギーメロン大学 ソフトウェア工学研究所が拡張した『Goal-Driven Software Measurement』

(c) 独フラウンホーファー研究所 (IESE) が提唱するGQM+Strategies

の3つの手法の情報を入手することができた。

GQM手法はソフトウェア工学における計測の枠組みおよびモデル化手法で、ゴール層 (G: 概念レベル) クエスチョン層 (Q: 運用レベル) メトリクス層 (M: 定量レベル) の三つの層に定義し、何のために測定するのか、結果をどう解釈するかなどを明確にした上で実施するという手法である。

(a)はGQM手法の最も基本となるオリジナルの手法である。

(b)は(a)を拡張したもので、「メンタルモデルを活用した要因分析」「Indicator (表示方法)」追加が大きな特徴である。図1はカーネギーメロン大学 ソフトウェア工学研究所が拡張した手法 (以降CMU版GQMと記す) の作業手順を示したフロー図である。

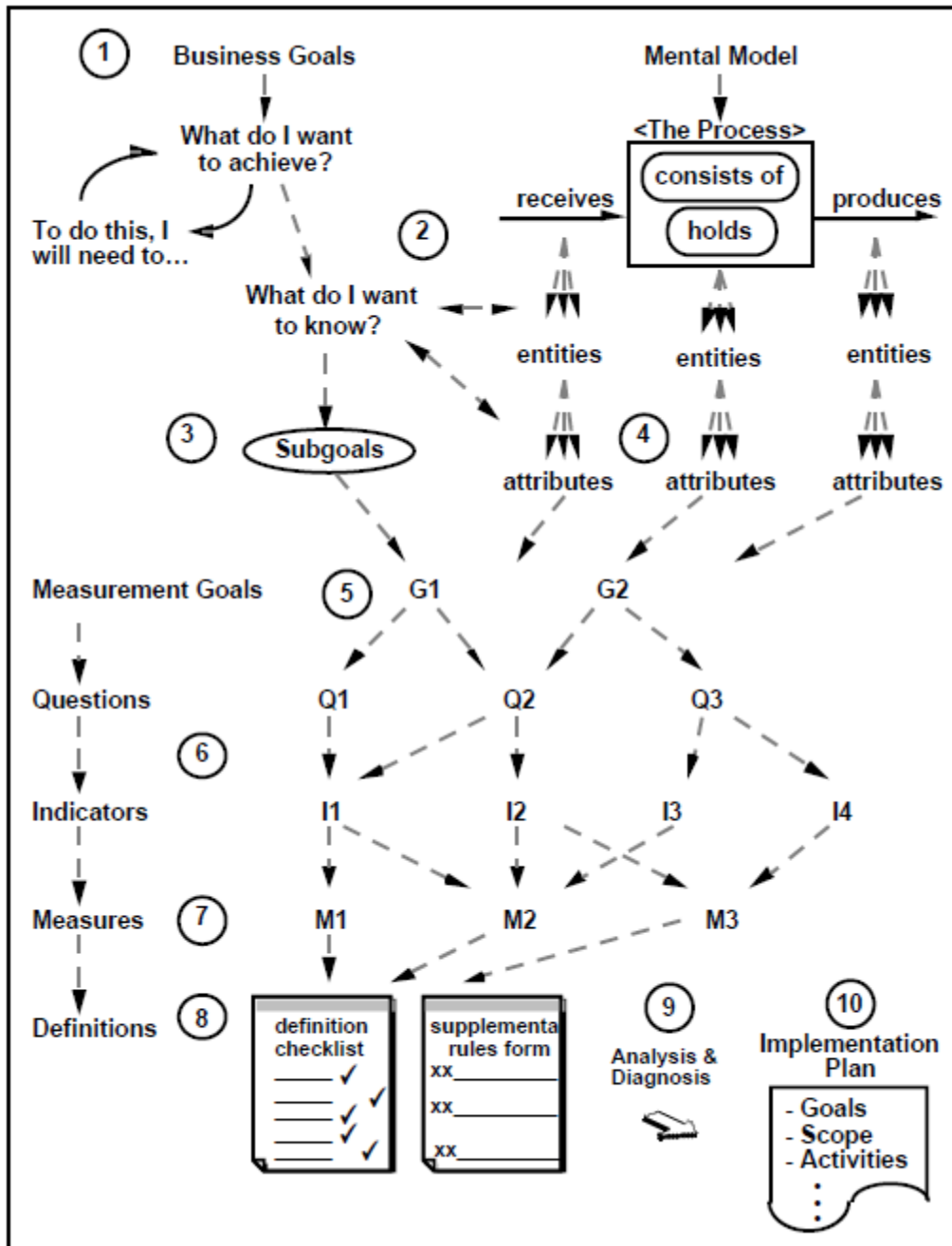


図1. CMU版GQMの手順フロー

CMU版GQMでは、図1の①～④の作業が拡張されている。まず①Business Goal(組織目標)を明確にし、②「組織目標を実現するために達成しなければならないことは何か」また「そのためには何が必要か」ということを繰り返す事で課題や施策を抽出する。またメンタルモデル(各人の頭の中にあるプロセスイメージ)を活用し、目標に関係しそうなエンティティ(プロセス、成果物、リソース等)やその属性(アトリビュート)を抽出する事で課題や施策を抽出する。その中からゴール達成に影響するもの(CSF: Critical Success Factor 重要成功要因)を選定し優先順位をつけて③サブゴールを定義する。④はサブゴールの定義後エンティティと属性を改良する工程である。⑤以降の作業は通常のGQM手法にあたるが、GQMのQuestionとMetricsの間にIndicatorを持たせることでGoal、Questionがより明確になり、且つ改善に結びつくMetricsの定義をスムーズにする。

(c)も(a)をベースに拡張したもので「組織の経営レベルの目標」「目標を達成するための戦略」「実務レベルの個別戦

略(例：IT化)」の三つの関係の整合性を取る事を実現した手法である。IT化の背景や期待される成果について経営レベルと共有にする事に重きを置いた手法で、戦略を決定した後のプロセスはオリジナルの GQM とほぼ同じである。特徴としては「事実/仮定による戦略抽出」「基幹システムの構築に適している事」である。表 1 に (a)～(c)の手法を比較した表を示す。

表 1. GQM の比較表

	(a) オリジナルGQM	(b) CMU版GQM	(c) GQM+Strategies
適用範囲	全般	全般	基幹システムの構築に特化
事実/仮定による戦略抽出	×	×	○
メンタルモデルによる要因抽出	×	○	×
Indicator	×	○	×

これらの3つの手法を比較検討した結果、(b)のCMU版GQM手法は、以下の効果が期待できるため採用することにした。

- ・メンタルモデルを活用することで明文化されていないプロセスや成果物及びその属性から背後に隠れて通常では気づけない要因を抽出することができる。
- ・Indicatorを作成することで測定結果が視覚的に解釈できるので、目標と測定の整合性を確認することができる。
- ・Indicatorそのものが結果の評価として活用できるため評価報告書も改善される

3. 評価の内容

1. で述べた課題を解決し組織目標を達成するためにCMU版GQM手法でメンタルモデルを作成した。図2は抽出したエンティティを示したものである。

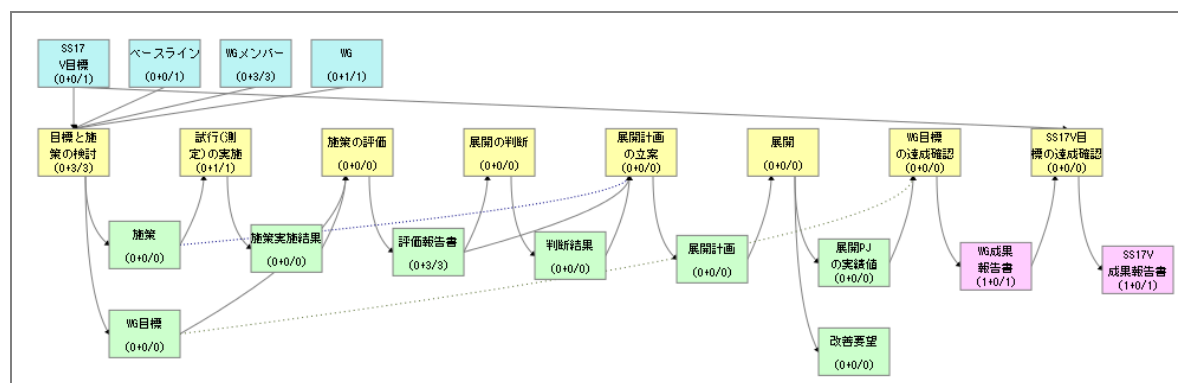


図 2. 組織目標達成のメンタルモデル (エンティティ)

次に図 2 で示したエンティティに関係する属性と質問を抽出し、目標達成に影響する CSF を選定した。結果を表 2 に示す。

表 2. ゴールに影響する属性 (CSF)

No.	ID	Entity	Attribute	Mark	Marked Question related to Business Goal
1: 目標と施策の検討					
1	I.3	WGメンバー	WGの対象工程を担当	○	WGの対象工程を現在担当しているか?
2	I.3	WGメンバー	なぜなぜ分析能力	○	根本原因の追究はできるか?
3	I.3	WGメンバー	GQM能力	○	評価に必要なメトリクスの定義ができるか?
4	I.4	WG	毎回参加者率	○	参加率の低いメンバーはいないか?
5	P.1	目標と施策の検討	要因抽出の妥当性	○	目標達成のための最大成功要因(CSF)が抽出できたか?
6	P.1	目標と施策の検討	成功要因抽出量	○	成功要因はどの程度抽出できたか?
7	P.1	目標と施策の検討	施策の妥当性	○	目標に対する正しい施策が立案できたか?
2: 試行(測定)の実施					
8	P.2	試行(測定)の実施	測定の合目的性	○	評価に必要なデータがどの程度測定されたか?
3: 施策の評価					
9	O.4	評価報告書	ビジネスゴール達成評価	○	ビジネスゴールに対する答えは記述されているか?
10	O.4	評価報告書	施策評価	○	施策に対する評価ができているか?
11	O.4	評価報告書	品質	○	評価報告書の品質は改善されているか?
7: WG目標の達成確認					
12	O.9	WG成果報告書	WG目標達成度合い	KGI	WG目標は達成したか?
8: SS17V目標の達成確認					
13	O.10	SS17V成果報告書	SS17V達成度合い	KGI	SS17Vは達成したか?

表 2 から特に期待する No 6、7、11 について評価することにした。また全社展開するには GQM 手法を実施する工数も評価する必要がある為「GQM 手法の習得にかかる工数」「GQM を実施にかかる工数」についても評価することにした。

- (a) 成功要因はどの程度抽出できるのか (表 3 の No.6)
- (b) 目標に対する正しい施策が立案できるのか (表 3 の No.7)
- (c) 評価報告書がどのように改善されるのか (恣意性の排除) (表 3 の No.11)
- (d) GQM 手法の習得にどれくらい時間がかかるのか
- (e) GQM 手法の実施にどれくらい時間がかかるのか

4. 試行方法

4. 1 試行計画

2015 年 4 月から各 WG で試行を開始するべく 2015 年 3 月～4 月に『Goal-Driven Software Measurement』の内容をベースとした研修を実施することにした。また GQM 作図用のツールも作成し、各 WG にツールを使った作成手順を OJT 形式で説明することにした。

4. 2 評価内容

- (a) 成功要因はどの程度抽出できるのか
WG が作成したメンタルモデルの CSF 数を計測 (ツールが自動計測)
- (b) 目標に対する正しい施策が立案できるのか
WG が立案した施策とその結果を評価
- (c) 評価報告書がどのように改善されるのか
評価報告書を作成し評価
- (d) GQM 手法の習得にどれくらい時間がかかるのか
WG メンバーが手法を習得するまでにかかった時間を評価
- (e) GQM 手法の実施にどれくらい時間がかかるのか

作成にかかった時間を WG メンバーから収集し評価

5. 試行結果と評価（考察）

計画どおり 2015 年 4 月から各 WG で試行を開始した。表 3 に試行実績を示す。

表 3. 試行実績

項目名		実績値	補足
研修	開催数	2回	・ 1時間/回 ・ 2015年3月、4月に開催
	受講人数	71名	
要因分析 (上流工程)	実施WG数	5チーム	
	作成テーマ数 (*1)	29件	
GQM (下流工程)	実施WG数	6チーム	
	作成テーマ数	13件	

(*1) 作成テーマ数：ビジネスゴールを検討できるサイズまでブレイクダウンし個人で作成した文書数

5. 1 成功要因の抽出

(1) 試行結果

表 4 に 4 つの WG で実施した要因分析（CMU 版 GQM の上流工程）の結果を示す。属性数はメンタルモデルから抽出したエンティティの属性の数で、重要な属性数はその中で CSF として抽出された属性数である。

尚、WG 会議ではメンタルモデルを共有する為通常の会議では気づけない要素に気づく事ができた。また視野が広がることで議論が活発になるという効果もあった。

表 4. 要因抽出

テーマ数	属性数	重要な属性数	時間（分/件）		テーマあたりの属性数	
			時間/属性数	時間/重要な属性数	属性数/テーマ	重要な属性数/テーマ
29	885	439	3.0	6.0	31	15

(2) 評価（考察）

一文書あたり属性を 31 件、重要な属性を 15 件も抽出できている。従来は一つの課題で 1 ～ 2 件程度の要因を決めて対策を立案していたので、当手法を使用することで多くの要因を抽出できたと考える。また一つの属性を抽出するのに 3 分/件、重要な属性も 6 分/件（いずれも加重平均値）と効率的に要因抽出できていると考える。

5. 2 目標に対する施策立案

(1) 試行結果

GQM 手法を使うことにより、いくつかの施策を立案できたが、まだ実施していない為妥当性の評価はできていない。

(2) 評価（考察）

定性的であるが、派生開発のコスト削減というテーマに対しては、本手法では母体プログラムの属性が多く上げられた。従来はプロセスや開発者の経験に注目していた（母体プログラムにはあまり注目していなかった）が本手法によって視野が広がりより重要な要因に対する対策が打てるようになったと考える。

5. 3 評価報告書の品質

(1) 試行結果

GQM ツリーから機械的に評価報告書を作成した。図3にGQM ツリーと評価報告書のサンプルを示す。該当のWGメンバーとレビューを実施した結果特に不足している項目はみつからなかった。

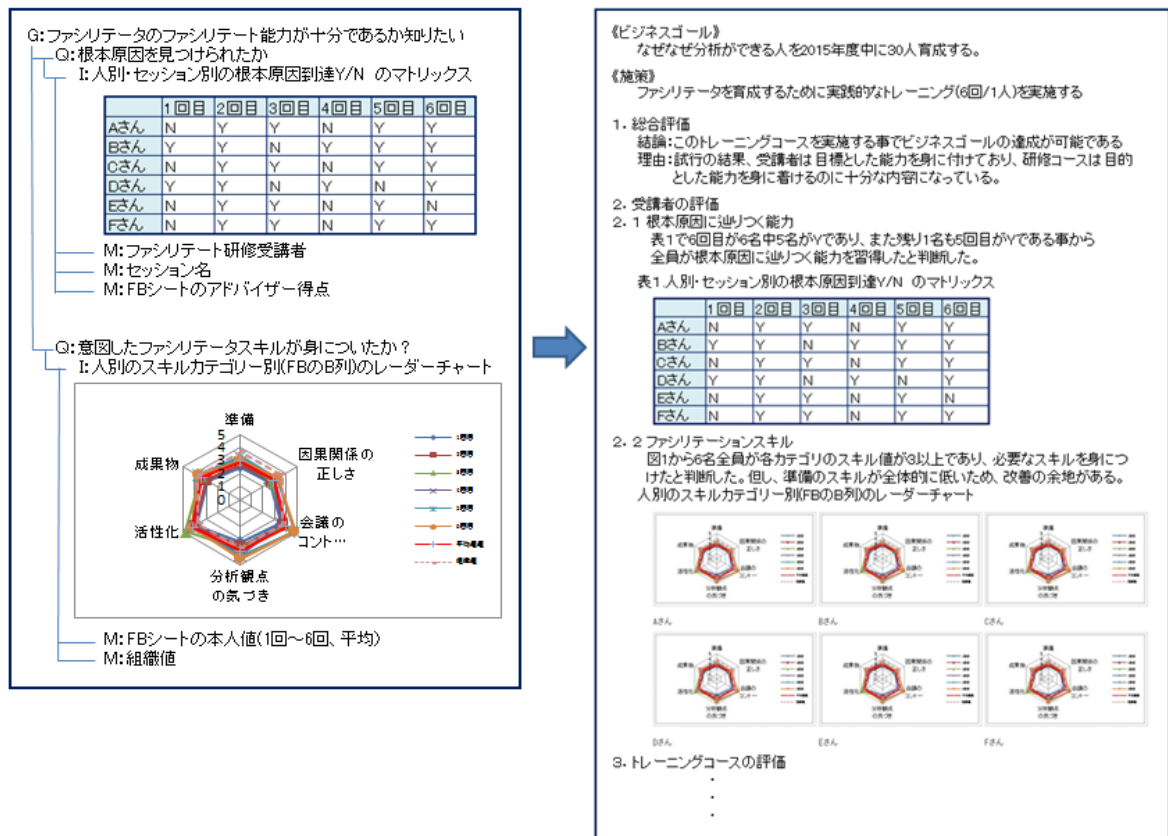


図3. GQM ツリーと評価報告書のサンプル

(2) 評価（考察）

GQM ツリー作成時にゴールに対する質問やその答え（解釈）が十分検討されているため、その時点で既にある程度の品質が確保できている。また Goal、Question、Indicator がそのまま評価報告書に使用できる為、評価報告書の作成時間は大幅に短縮できると考える

5. 4 GQM 手法の習得時間

(1) 試行結果

多くの WG メンバーは研修 1 時間と OJT(WG 会議内の試行)2 時間で、基本的な GQM 図を作成できるようになった。

(2) 評価（考察）

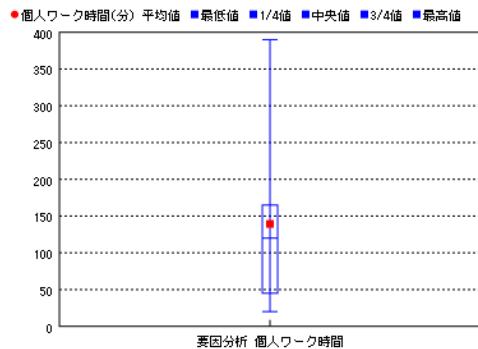
GQM 手法の習得時間は個人の経験やスキルにもよるが、研修 1 時間と OJT(WG 会議内の試行)2 時間

の計 3 時間で、基本的な GQM 図を作成できる。基本的な実施手順は 3 時間で習得可能と判断した。

5. 5 GQM 作成時間

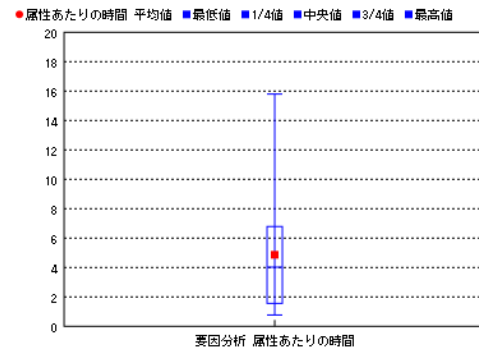
(1) 試行結果

要因分析（CMU 版 GQM の上流工程）の作成時間とそのバラツキを図 4 及び図 5 に示す。



件数	合計値	平均値	最低値	1/4値	中央値	4/3値	最高値
20	2785	139.3	20	45	120	165	390

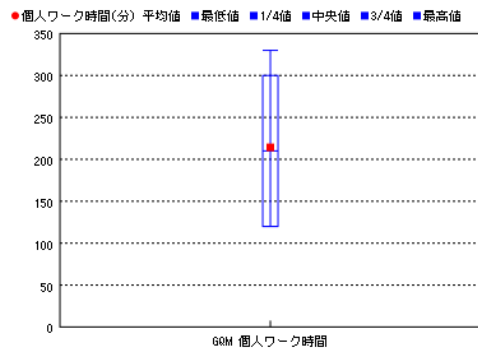
図 4. 要因分析 個人ワーク時間



件数	平均値	最低値	1/4値	中央値	4/3値	最高値
20	4.86	0.77	1.56	4.03	6.79	15.80

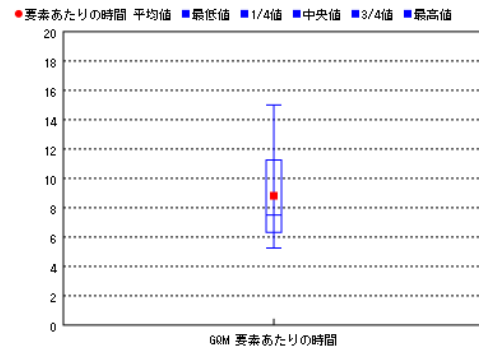
図 5. 要因分析 属性あたりの時間

GQM（CMU 版 GQM の下流工程）の作成時間とそのバラツキを図 6 及び図 7 に示す。



件数	合計値	平均値	最低値	1/4値	中央値	4/3値	最高値
7	1500	214.3	120	120	210	300	330

図 6. GQM 個人ワーク時間



件数	平均値	最低値	1/4値	中央値	4/3値	最高値
7	8.81	5.25	6.32	7.50	11.25	15.00

図 7. GQM 要素あたりの時間

(2) 評価（考察）

要因分析(上流)の作成時間は個人のレベルにより多少ばらつきがある(最小 20 分 最大 6 時間半)。まだ記述標準やノウハウが十分でない為、ツールに慣れない、何をすればいいかわからない等で時間がかかっている人もいるが平均約 140 分でメンバーで議論できるレベルの成果物が作成できている。一属性あたりで見ると平均 4.86 分/件で許容範囲と考える。

GQM 作成（下流）は 平均 214.3 分、一属性あたり 8.81 分/件で要因分析より時間がかかっている。これは実際の測定や評価をするための細かな定義が必要である為妥当な工数と考えている。

6. 結論

冒頭に述べた通り、現状は以下の課題があった。

(a) 目標値の妥当性が低い

- (b) 施策の妥当性が低い
- (c) 試行が目的になっている（目標と施策がずれている）
- (d) 場当たりの評価になっている

(b)の評価はまだ途中であるが、(a)(c)(d) に効果があることが確認できた。また手法の展開に関して教育コスト及び実施コストを評価した結果、特に問題となるコストではなかった。GQMの実施には時間がかかるものの評価報告書の作成時間が大幅削減できるのでトータルとして増加工数は少ないと考える。以上の結果、CMU版GQMはプロセス改善活動に有効な手法であると判断し組織展開することにした。

7. 今後の課題（導入への課題）

GQM ツリーの作成、及びレビューやディスカッションに時間がかかるので効率化が必要と考える。数か月の試行でかなりノウハウが溜まったので、ツールや記述標準(ガイドライン)を改良し作業効率の向上を目指す。

8. 参考文献（資料）

- [1] Victor R.Basili,Gianluigi Caldiera,H.Dieter Rombach,“Goal,Question, Metric Paradigm, Encyclopedia of Soft-ware Engineering”, 1994
- [2] Robert E.Park, Wolfhart B.Goethert, William A.Florac,“Goal-Driven Software Measurement A Guidebook”1996
- [3] IPA,“GQM+StrategiesR ワークショップ教材”,2013
- [4] Peter M.Senge,“最強組織の法則—新時代のチームワークとは何か”,徳間書店,1995
- [5] 伊沢 武史,“GQMを用いたメトリクス定義と測定・分析システムの構築”,SPI Japan 2014

3C3 「自律型プロジェクトチームへの変革アプローチ事例」 安達賢二(HBA)

<タイトル>

自律型プロジェクトチームへの変革アプローチ事例

<サブタイトル>

チームの価値観変容を重視し、問題モデリングを活用した SaPID 流プロセス改善アプローチ

<発表者>

氏名（ふりがな）： 安達 賢二（あだち けんじ）

所属：株式会社 HBA

<共同執筆者>

（なし）

<要旨>

人間やチーム・組織の行動や成果は、「価値観→考え方→行動→結果・成果」の流れで説明できると思います。そこで、チームの価値観変容を重視したアプローチを採用することで、メンバーの考え方や行動が変わり、最終的にプロセスや成果が向上することを目指した自律型プロセス改善手法 SaPID(*1)の適用事例を報告します。

チームのメンバー各自がそれぞれバラバラに個別の問題を抱え、解決していく状況から、問題モデリング(*2)を活用してチームワークを重視する価値観に変容し、どのような段階を経て自律したチームに変わっていくのか、その過程でどのような課題をクリアしていったのか、メンバーがやらされ感やストレスを感じずに変容していくためのポイントなどを共有したいと思います。

*1：SaPID = “Systems approach based Software Process Improvement methoD”の略で、当事者自らが解決すべき問題を現実的に解決しながら段階的・継続的にゴールを目指す手法

*2：問題モデリング（広義）とは、チームや組織に存在するたくさんの問題と、それらがどのように関連して何が起きているのか（因果関係）をわかりやすく表現し関係者全員が問題を（最終的には全体構造と、個別詳細の両面で）把握・理解し、納得することで関係者全員に“問題発見・解決・改善の当事者”になってもらうことを目指す過程と結果の総称。その過程を「問題モデリングプロセス」、問題の表現方法を「問題モデリング（狭義）」と呼ぶ。

<キーワード>

自律型プロジェクトチーム,問題モデリング,チームとしての大事な価値観,問題定義,情報共有,相互コミュニケーション,ふりかえり,問題構造図,試行,段階導入,価値観変容,運営役育成,リーダーによる統率とコーディネート,改善実践の業務との一体化

<想定する聴衆>

チームによる自律運営を目指す方。部署レベルで自律運営のできるチームを作り出し、広げていきたいマネージャや管理者の方。自律型プロセス改善手法 SaPID を状況に応じてどのように適用し、改善を進めていくのがよいか、具体的な事例を知りたい方。

<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（投稿時点で実践中（一部結果あり）／発表時には最終結果が出ている予定です）

<発表内容>

1.背景

首都圏の顧客から委託された組込み系システムに対するシステムテスト分析・設計・実行を主な業務として
いる部署（社員 20 名、協力会社 50 名程度）が今回の対象です。

組込み製品群のさまざまなバージョン、種類に対するテスト業務を請け負うプロジェクトチーム（1 チーム 5～
20 名程度）が 5～7 チームあります。（業務の状況に応じてチーム数、人数は変化します）

リーダ、メンバーはみな真面目で依頼された作業は実直にこなすため、ある程度の成果は上がるものの、難
易度の高い案件や制約条件が厳しい案件には弱い傾向でした。よく言えば真面目で実直、悪く言えば受
身で消極的。また、顧客からは「もっと積極的に提案を！」「スキルアップを含めた説得力のある改善を！」と
の要望が上がっているものの、急に発生する大きな問題への対応で手が回らず、急場しのぎを繰り返してい
ました。その打開を目指して私が自律型プロジェクトチームへの変革アプローチを提案し、その適用支援を準
備、開始しました。

2.改善前の状態

試行を兼ねて先行導入対象とした 2 チームは「テスト実行」が主な業務です。テスト実行では、必要なテスト
環境を整えてテストを実施し、結果を取りまとめる。その過程で関係する開発ドキュメントや過去バグ情報な
どの調査、顧客窓口や開発担当者、テスト設計担当者などへの確認・調整などを行っています。

各チームでは、いつもさまざまな問題と残業が常態化しており、メンバーは自らの担当作業に没頭、コミュニケ
ーションが少なく、他のメンバーに問題が発生していることもわからず、自らの作業で問題が発生すると残業・
休出し自分だけで何とかする（が、大きな問題になると全員が駆り出される）、チーム共通の問題があつて
も「誰かが解決するのだろう」的に放置し、リーダだけが四苦八苦するのがあたり前な状態でした。

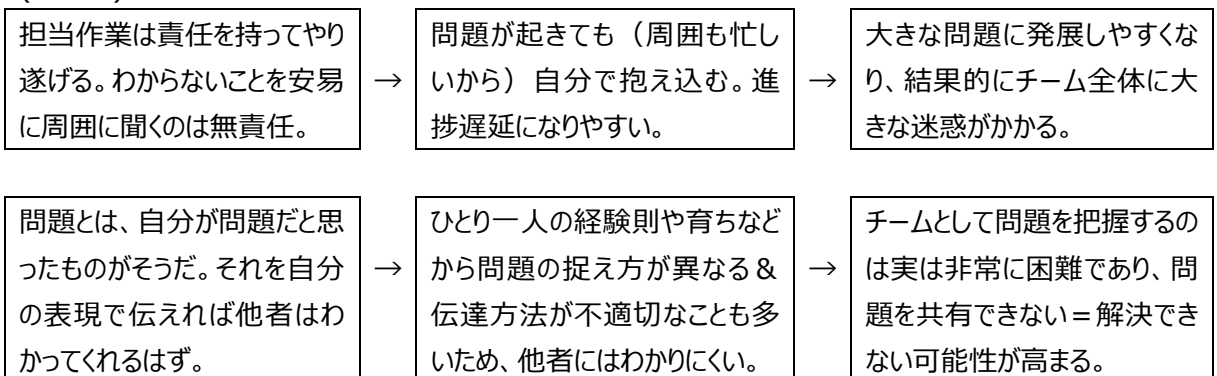
みな真面目で依頼された作業は実直にこなすため、ある程度の成果は上がるものの、難易度の高い案件や
制約条件が厳しい案件には弱い傾向でした。

今回のアプローチにより、誰かに言われなくてもチームとして問題を早期に発見・解決する、必要な改善を明
確化し、チームワークよく実践できるようになり、顧客にもアピールできるようになることを期待しています。

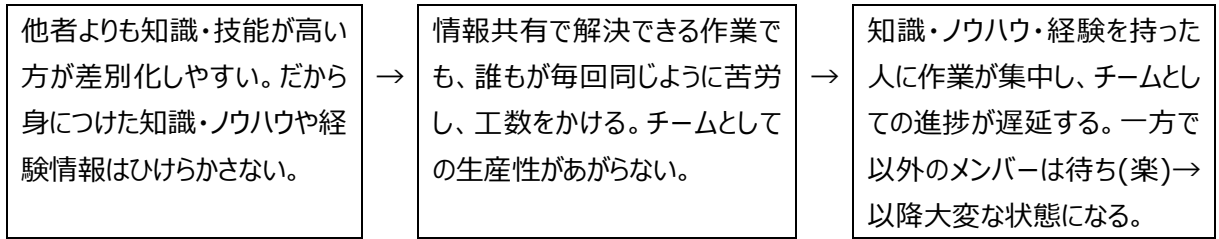
3.改善前の状態をもたらした原因（因果関係）

改善前の状態は、主に以下のような価値観や考え方・行動から成り立っていると考えました。

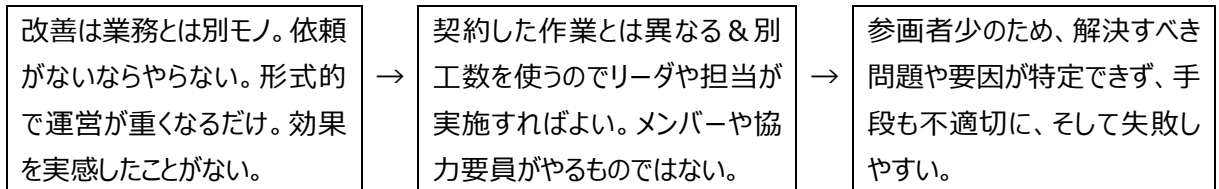
(要因 1)問題の早期発見・解決を阻害する価値観・行動・その結果



(要因 2) 情報共有を阻害する価値観・行動・その結果



(要因 3) 効果的な改善実践を阻害する価値観・行動・その結果



これらの要因や負の連鎖は、対象プロジェクトの状況を 3 か月間（2015.3～5）にわたり観察した結果に基づき、仮説・推論により導きました。最終的には 2 人のプロジェクトリーダーに確認した結果でもあります。以上の内容は特別なものではなく、どこのプロジェクトでもありがちなものと認識しています。

4. 計画した変更内容

	変更内容（問題モデリング方法の変化）	選定理由
(1)	チームとして「問題」を定義し共有する。	・何を問題とするのか、についてチームで共有し、メンバー各自が問題をわかりやすく提示できるようになるため。
(2)	日次情報共有 メンバーそれぞれの日報コメント欄に、その日の「よかったこと、うまくいったこと」「困り事や問題点」「次にどうするか」を記載し、提出してもらう。 リーダーが日々のメンバーコメントに個別フィードバックコメントを入れサマリしたものをチーム全員で共有する。	・相互コミュニケーションを促進し、メンバーが何を考えているのか、どのようなことが起きているのかを把握するため。 ・個別の問題に対して知っている人がいれればすぐに情報を提供し、解決してもらうため。（それがよいことだと認識してもらうためでもある）
(3)	週次ふりかえりミーティング開催 メンバー全員が集合して一週間のコメントなどからその前週の代表的な「よかったこと、うまくいったこと」「困り事や問題点」「次にどうするか」を議論し、まとめる。その際に、支援者より、その段階に必要な話題を提供し、意見交換や議論・検討などにより一つの仮説や結論を導く。	・このような運営に変えてうまくいったこと（例：個別問題の解決）とその意味・意義を確認し、実感するため。 ・個別問題の改善を検討（翌週に実践）するため。 ・運営のしやすさ、しにくさを把握し、より費用対効果が高くなるように改善するため。 ・チームとしての大事な価値観を模索・共有し、次の活動に活かして実践してもらうため。

(4)	<p>月次実績情報によるふりかえり開催 メンバー全員が集合して直近一か月の情報共有・ふりかえり運営の実績情報から「できるようになったこと」「まだできていないこと・課題」「今後どうするか」を議論し、まとめる。また、一か月に上がった問題要素の構造分析結果（問題構造図）から、チームとして解決すべき問題を特定し、どのように解決するのかを検討・決定する。</p> <p>その際に、支援者より、その段階に必要な話題を提供し、意見交換や議論・検討などにより一つの仮説や結論を導く。</p>	<p>・このような運営に変えてうまくいったこと（例：個別問題の改善）とその意味・意義を確認し、実感し、さらに自分事として認識してもらうため。</p> <p>・チーム内問題の中から改善すべき事項を特定し、改善を検討（以降実践）するため。</p> <p>・チームとしての大事な価値観を模索・共有し、次の活動に活かして実践してもらうため。</p>
-----	--	--

5.変更の実現方法

■ 試行を含めた段階導入

まずは先行して2チームで試行。その結果から全体に展開する予定。

（今回の事例報告は試行分のみ）

■ 段階的な課題設定と問題モデリング方法の変化：

まずはチーム内部の「個別問題の早期発見・解決」ができるように。（上記表(2)）

次に、それらに含めて「個別問題の改善」を実践できるように。（上記表(3)）

最終的には、さらに「チームとしてのプロセスの改善」も併せて実践できるように。（上記表(4)）

■ チームとしての価値観変容に向けて：

どうやるのか、よりも、なぜそうするのがよいのか、チームにとってどのような価値があるのかを中心に考えてもらい、理解していくアプローチとした。

適宜、簡潔に外部関連情報の紹介や現状との類似性なども補足し、解説する。

■ 運営役育成方法：

当初は支援役がポイントを解説したうえで実践し、ノウハウを共有する。

支援役が2～3回実践後、実践役をリーダーに交代し、その実施を支援する見守りモードに変える。

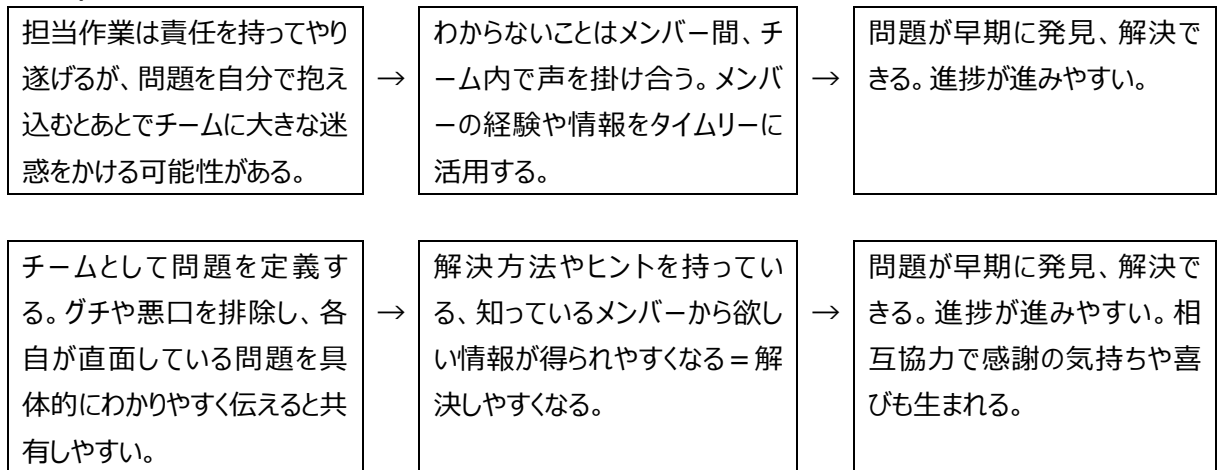
毎回終了後に簡単に実施ポイントごとの評価と次の課題を伝える。

6.変更後の状態や改善効果

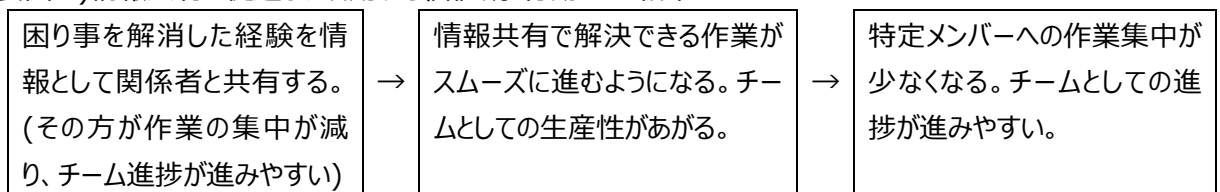
以下のような変化があること、結果となることを目指して活動中です。

最終的には実践結果から、これらを裏付ける結果やヒアリング結果と総括を提供する予定です。

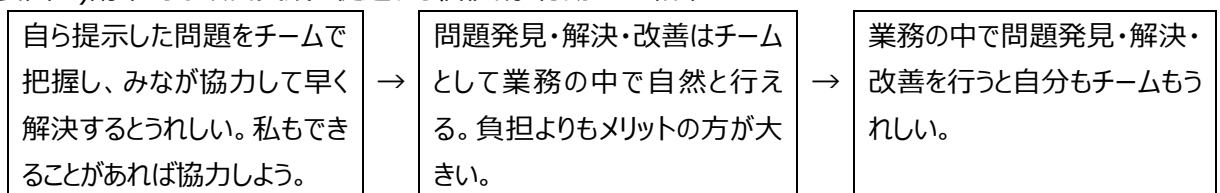
(要因 1)問題の早期発見・解決に対するチームとして適切な価値観・行動・その結果



(要因 2)情報共有を促進し、活用する価値観・行動・その結果



(要因 3)効果的な改善実践を促進する価値観・行動・その結果



リーダーが自らこれらを統率、コーディネートして実践し、結果を獲得する。

問題発見・解決、そして必要な改善実践が業務と一体化していて、メンバーもそれがあたり前だと思っている。

7.改善活動の妥当性確認

「1」で記述した「背景」を踏まえて、活動全体を分析し、変更内容および実現方法の妥当性、費用対効果、残存課題、副次効果などを確認した結果を報告する予定です。