SPI Japan2017 発表概要集

・セッション順番 ＞ 発表順番に記載しています。

1A1「再発防止のための問題・課題分析と特許分析ツールの出会い」藤山晃治（パナソニック）....3
1A2「「あるある診断ツール」による課題の見える化と収集データ分析」室谷隆（TIS）....7
1A3「文書作成支援ツールによる組織開発力の強化」中村伸裕（住友電工情報システム）....11
1A4「ユーザワーキングをプロセスに組み込んだ品質向上」山中美圭代（ダイキン情報システム）....21
1B1「医療現場の IT 製品開発でスクラム・オブ・スクラム -組織のマインドを変革する！-」清水弘毅（オリンパス）....26
1B2「Agile 開発フレームワークを応用した効率的な開発ナレッジの獲得」林健吾（デンソー）....31
1B3「アジャイル開発におけるプロセス改善事例 ～楽しく・早く・確実に～」中野安美（ニッセイ情報テクノロジー）....36
1B4「アジャイル推進活動をアジャイルにやってみた」伊藤裕子（東芝）....41
1C1「異なる開発組織における継続的統合に向けた Queuing 統合の提案と適用」白木徹（デンソー）....45
1C2「高信頼アーキテクチャ設計手法 ATAM の実践」藤原啓一（三菱スペース・ソフトウエア）....50
1C3「プログラム設計要否判定による工数削減」丹羽郁美（住友電工情報システム）....54
1C4「製品ラインナップと開発制約に基づく丁度良い仕様の開発方法の提案」髙橋拓未（デンソー）....61
1D1「CMMI を軸としたニアショア開発における品質改善活動」井上靖章（クオリサイトテクノロジーズ）....65
1D2「IT サービスマネジメントの観点を考慮した運用保守プロセスの評価と改善」中村英恵（NTT データ）....68
1D3「CMMI を活用した社内版プロセス評価モデル構築による全社品質・生産性向上への貢献」柳田礼子（NEC）....71
1D4「パーソナルソフトウェアプロセスに基づく実践的ソフトウェアエンジニアリング教育の効果と自己改善活動におけるインストラクタの役割」梅田政信（九州工業大学）....74
2A1「課題管理からアプローチするトラブルプロジェクトの立て直し」飯田秀樹（オープンストリーム）....78
2A2「デザイン思考を活用したプロセス改善」服部悦子（住友電工情報システム）....82
2A3「SEPG 活動の立ち上げと、組織定着への取り組み」清水崇司（ニコンイメージングシステムズ）....91
2B1「WHY に重点を置いた人材育成」片山泰司（オムロン ソーシアルソリューションズ）....95
2B2「人に満足を与えるものづくり・コトづくりのマインド改革」島林大祐（富士通）....98
2B3「ヘルスケア商品における「自律」を主眼に置いたプロセス改善」伊達渡（オムロン ヘルスケア）....102
2C1「IoT に欠かせない BLE 通信のテスト自動化によるテストプロセス改善」伊藤卓也（オムロン ヘルスケア）....105
2C2「UT 仕様書自動出力システムによる UT 工数削減の取組」野尻優輝（住友電工情報システム）....110
2C3「システムテスト自動化の普及・展開に向けた取り組み」小笠原秀人（東芝）....115
2D1「社内エキスパートの育成による、ソフトウェアプロダクトライン(Software Product Lines)の全社展開」丹羽徹（オムロン）....122
2D2「プロダクトライン開発における複数製品導出の同時並行開発方法の提案」荒木邦彦（デンソー）....126
2D3「自動車ボディ系システムにおけるプロダクトライン開発の導入と実践」浅野雅樹（アイシン精機）....131
3A1「急成長プロダクトの Dev&Ops で生じる悪循環とその解決策」横山翔（リクルートテクノロジーズ）....136
3A2「DevOps でリードタイムを８ヶ月 から最短１週間まで短縮！！」安藤寿之（NEC ソリューションイノベータ）....145
3A3「基幹システムの開発・保守における機械学習の適用検討とその評価」陣内孝司（住友電工情報システム）....148
3B1「自分事化影響要因に着目した中期経営計画立案・展開への共創アプローチ[現状分析～計画立案編]」安達賢二（HBA）....155
3B2「GQM を拡張した WG 活動の短期化」山邊人美（住友電工情報システム）....166
3B3「「かんばん」から始まった改善と制度化のジレンマ」竹縄俊政（日新システムズ）....174

3C1「システムアーキテクトと取り組む品質管理」福良智明（オープンストリーム）..... 180
3C2「品質目標と工程別活動内容シートの活用事例の紹介」相澤武（インテック）..... 183
3C3「「なぜ」に頼らない「なぜなぜ分析」のすすめ」渡辺聡美（富士通エフ・アイ・ピー）..... 187

1A1「再発防止のための問題・課題分析と特許分析ツールの出会い」藤山晃治（パナソニック)

<タイトル>

再発防止のための問題・課題分析と特許分析ツールの出会い

<サブタイトル>

～言葉が導くプロセス改善への糸口～

<発表者>

氏名（ふりがな）：藤山　晃治（ふじやま　こうじ）
所属：パナソニック(株)　オートモティーブ＆インダストリアルシステムズ社　技術本部　システム技術開発部　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）：　なし
所属：

<主張したい点>

問題解決管理プロセスの実施に当たり、進行中のプロジェクトで発生した問題・課題を適切に解決に導くことに加え、どのようなプロセスで何が原因で問題が発生したかなどに関する再発防止のための分析が重要である。しかし、発生件数ベースの分析だけでは、どこを変えればいいかは分かるが、何をどのように変えれば改善できるのかまでは知ることが難しい。本発表では、問題・課題に関して具体的に記載されている文章の分析を通して知見を収集し、プロセス改善や再発防止への手がかりを示す。その際に、特許分析・マップ作成ツール「パテントマップ EXZ」を活用する。特に、通常の表計算ソフトでは難しい言葉の分析を行うことで、改善を行うため視点がどのように具体化されていくかを実例とともに紹介する。

<キーワード>

課題、問題、問題解決管理、分析、パテントマップ EXZ、プロセス改善、再発防止

<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、プロセス改善担当者、SQA など

<活動時期>

2016 年 10 月～2017 年 3 月

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
■ 改善活動を実施したが、結果はまだ明確ではない段階
□ 改善活動の結果が明確になっている段階
□ その他（　　）

<発表内容>

1.背景

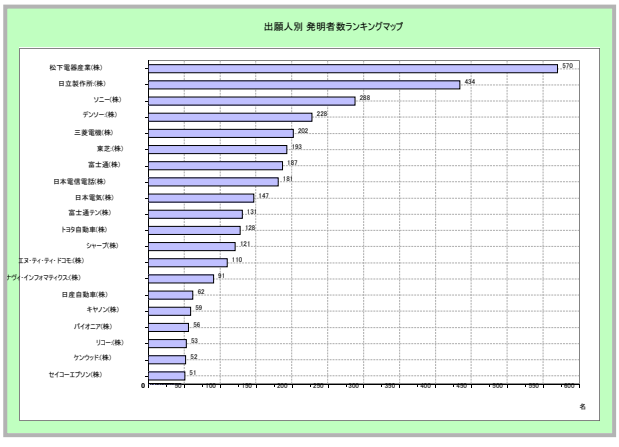
プロジェクトを遂行する上で、重要なプロセスに「問題解決管理プロセス」がある。このプロセスでは、認識した課題や発生した問題を適切に解決に導くことが求められるが、一方で、発生した問題・課題の再発を防止するために、それらを分析し後続の開発に反映させていくことも等しく重要である。大抵の場合、蓄積された問題・課題データは、発生件数を中心に分析され改善活動方針に反映される。これらの分析は表計算ソフトで処理された後、項目別に件数が視覚化され、「要件分析プロセスでの要件定義ミスによる問題発生件数が多いため、要件分析プロセスを強化する」というような施策へ落とし込まれる。しかしながら、そのレベル終わっているケースが多く、現場の技術者が積極的に動けるような具体策まで落とし込めていない印象を筆者は持っていた。従って、せっかく貴重なデータベースがあるのに、それを十分に活用できてない状況を変え、具体的な改善指針でもって現場の高品質な製品開発に貢献したいと考えていた。

2.改善したいこと

このように、改善したかったのは、「現場の技術者が実感・理解し、具体的に再発防止に意欲的に取り組めるようにしたい」という点である。筆者のように第三者的な立場でプロセス改善を進める場合は、プロセス定義書に書かれていることを遵守させるためにあるべき論を説いたり、仕組み構築を行ったりするが、効果が中長期的に表れるものが多い。そこに、即効的な方策を期待する現場の技術者との温度差が生まれることがある。よって、筆者は前者の必要性を認識したうえで、どうにかして現場の技術者が共感できるプロセス改善策や問題の再発防止策はないか、と長年考えていた。過去に同じような目的で問題が分析された報告書を目にしたことがあったが、発生件数の推移的な分析や比率分析が中心であり、具体策への落とし込みが不十分と感じた。同時に、貴重な情報がたくさん記載されているはずの、問題の発生状況の説明部分の包括的な分析が全く行われていないことも知った。それは、おそらく、表計算ソフトを使っても人手で行っても、ある程度短期間で効率的に分析結果が出せないことが原因ではないか、と考えた。文章を単語レベルに分解するという作業は、表計算ソフトが苦手とするものであり、しかも数百件を超えるようなデータでそれを効率的に行うことはまず不可能である。また、人手で解釈し読み解くにしても限界があり、まして報告された複数の問題同士から因果関係を探り出すことはできない。だからこそ、この問題・課題として記載された文章情報を読み解き、問題の文章に隠れた単語間の因果関係をあぶり出すことなどができれば、技術者が実感できる新たな具体的な改善の糸口が見えてくるのではないか、という結論に至り、その方法を模索した。

3.改善策を導き出した経緯

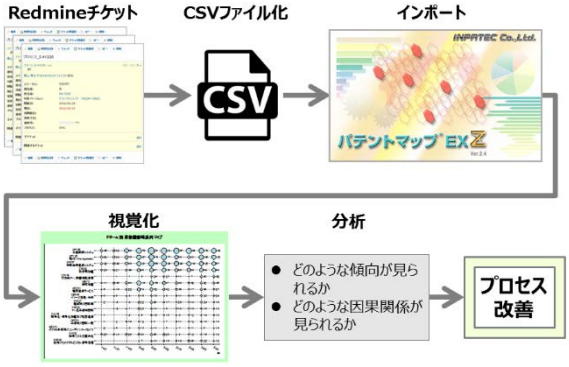
さて、筆者はかつて研究開発部門において研究テーマの戦略立案を行う部署に所属していたことがある。戦略立案のためには、特許分析が必要不可欠である。その際、特許部門から「パテントマップEXZ」[1]という特許情報分析・マップ作成ツールの紹介を受け、約２年間、本部内のさまざまな研究テーマに関する特許分析を本ツールで行った。このツールは図１に示すとおり、パテントマップ作成を支援するツールであり、大きな特長は表計算ソフトが持つ件数ベースでの分析およびグラフ化機能に加え、特許にある請求の範囲や要約、本文に記載されている文章情報を単語レベルで分割して管理し、単語の登場件数や単語間の関係のグラフ化が簡単に行える点である。しかも、特許が持つ言葉の揺らぎ(車両と車輌、太陽電池と太陽光発電、電気自動車と EV など)を統合でき、同じ意味を持つ別表現の単語をユーザ定義により統一したり関連付けたりして管理できる、独自の優れた特長も多数有している[2]。筆者はこのツールの優位性に痛く感銘を受けた。そして、現業務であるプロセス改善を行ううえで、このツールを使えば問題・課題として記載されている文章の分析を精細なレベルで行えるのではないか、という方法を思いついた。



【図 1】出願人ランキングマップ

4.改善策の内容

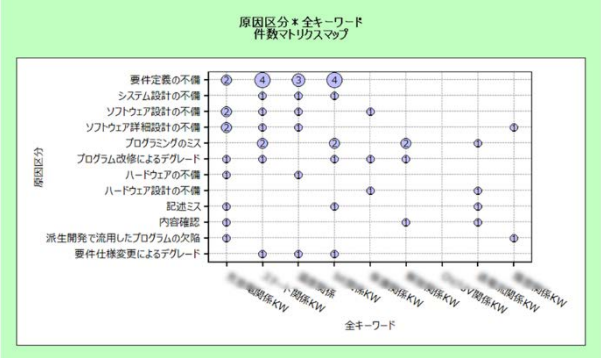
問題解決管理プロセスは、Redmine のようなチケット駆動型のツールを導入して実施される場合が多い。今回、ある部門(A 事業部とする)の Redmine 導入と試行的に Redmine を導入したプロジェクトの終了に伴い、蓄積データの分析の依頼を受け、3 章に記載した方法論に基づく分析を提案し、改善の方向性の指摘を行った。取り組みの流れを図 2 に示す。蓄積された問題・課題データは、CSV のような表形式のファイルのエクスポートできる。一旦、情報を表形式で保存できれば、パテントマップツールにインポートすることができ、分析の準備が整うため、A 事業部からは CSV 化された問題・課題データを頂いた。筆者が行った具体的活動は以下の 2 つである。



【図 2】取り組みの流れ

① パテントマップツールに読み込まれた課題・問題情報の分析

- 通常のチケット登録数の時系列データ分析、問題・課題の発生プロセスの件数ランキング、比率分析など。
- 今回の分析の特徴となる、問題・課題の記述に頻出する言葉と、その原因の因果関係特定。問題・課題表現の文章に含まれる言葉のランキング、言葉間の関係分析(セットで登場する言葉はあるか)など。
  - 図 3 は問題の原因区分と、問題の記述に登場した言葉の関係を示し、丸が大きい部分はその区分でその言葉が登場したチケットが多いことを示す。これにより、言葉によっては問題の原因区分が異なることが分かる。



【図 3】問題の原因区分と、登場した言葉の関係を示すマトリクスマップ

② 分析報告書の作成とプロセス改善方向の示唆

- 分析結果に、それを踏まえたマクロ視点でのプロセス改善の方向性と、技術者が納得できるミクロ視点での改善の方向性の示唆
  - ミクロ視点の例をとれば、要件定義の不備に頻出する言葉を特定し、再発防止のために、その言葉をどのように扱わなければならないかの知見を引き出し、再発防止策を練る。

5.改善策の実現方法

4 節で一例を示したが、分析により以下のような改善点が見えてきた。

- ある機能(言葉の分析より該当機能を特定)を許可/禁止する時に、不具合が多く発生している。
- ある機能はソフトウェア中心に、ある機能はハードウェアとソフトウェアの両方で不具合が発生している。
- ある機能に関しては、顧客側のテストで多く不具合が発見されている。
- ある機能は、特定の別機能とセットで問題が発生している。

これを元に、技術的なミクロな観点と、組織的なマクロな観点で改善策を立案した。

1. 【ミクロ観点】技術的な詳細に関する問題：設計ガイドラインやチェックリストへの反映
   特定の機能を実装する際に考慮すべきポイントは、技術者が具体的に設計に実装できるよう、設計ガイドラインやチェックリストへ記載した。
2. 【ミクロ観点】問題解決管理チケット登録方法のガイドライン作成
   ツールの優れた機能に依存することは可能であるが、より正確に状況を分析できるよう、チケットに登録する際の表現に関するガイドラインを作成、例えば、「○○の不具合」は、「○○が△△するときに□□できない」のような記述に変えるな

ど、推奨される表現などを準備し、現場の技術者が直接参考にできる形にした。

3. 【マクロ観点】より大きな、かつ抜本的な問題：具体的に組織開発プロセス改善内容を記載
   例えば、顧客テストで多く発生している機能に関しては、事業部でのテスト内容・方法との比較を行い、欠けているテスト項目を分析して開発標準へ追記した。また、最終段のテスト依存を避けるよう前段のテストで確認すべき項目を明確に記載した。
4. 【マクロ観点】プロジェクト終了報告時のインプット文書とする旨の組織開発標準への記載
   A 事業部では、プロジェクト終了時には、会議を開催しコストや日程といった項目に関する計画とのずれをふり返っていたが、今後は、この問題・課題分析報告を会議のインプット文書として、会議内で技術的な再発防止策を共有できるように組織の仕組みを変更した。

6.改善による変化や効果

1. 現場技術者の意識の変化
   A 事業部の現場の技術者は、進行中のプロジェクトにおける問題解決管理の重要性は理解していたが、再発防止のための活用に関しては十分に検討されていなかった。今回、注意するべき点が絞り込まれている分析を見ることで、単刀直入に対策を打てるということに気付き、再発防止意識が高まった。
2. 問題解決管理のチケット登録時の記載への配慮
   進行中のプロジェクトの問題解決に限れば、問題の記述に関しては関係者の理解が得られれば良い、という現場の理解であったが、再発防止目的にも使うために、①できるだけ具体的に記載する、②言葉を統一する、③発生プロセス、対応プロセスなどの情報を漏れなく入力する、といった点に注意して、チケット登録を行うようになった。
3. ミクロ視点の取り組みに基づく、マクロ視点の取り組みの理解醸成
   ミクロ視点の取り組みとマクロ視点の取り組みを連動させることで、現場の技術者も組織的なプロセス改善の意義を理解するようになった。

そのほか、SPI 担当者からは、この活動を含む全体的なプロセス改善の効果かどうかは分からないが、事業部として顧客流出不具合が減少傾向にあり、今後も継続していきたい、とのコメントもあった。

7.改善活動の妥当性確認

A 事業部からは、「第三者の視点から、特にプロジェクトの内情を知らない立場での分析」を依頼されたため、できるだけそうあるように努めたが、分析内容が的外れにならないよう途中何度か関係者に中間報告とディスカッションを行った。その結果、現場の技術者やプロセス改善担当者が要望する観点を盛り込み、納得のいく活動にすることができた。

A. 参考情報

[1]パテントマップ EXZ、インパテック株式会社 HP、https://www.inpatec.co.jp/
[2]「パテントマップ EXZ を用いた研究開発プロセスの効率化・最適化」、セミナー資料

1A2「「あるある診断ツール」による課題の見える化と収集データ分析」室谷隆（TIS)

<タイトル>

「あるある診断ツール」による課題の見える化と収集データ分析

<サブタイトル>

～保守/運用（エンハンスメント）の改善を楽に、楽しく（エン楽）

<発表者>

氏名（ふりがな）：室谷　隆

所属：　ＴＩＳ株式会社　生産革新本部　生産革新部　エンハンスメント革新室

<共同執筆者>

氏名（ふりがな）：

所属：

<主張したい点>

「あるある診断ツール」は

・だれでも簡単に（約３０分）

・現在の PJ 状態をチェックするだけで

改善課題を見える化できる、エクセルで作られた簡易アセスメント（セルフアセスメント）ツールである。

保守 PJ や、運用 PJ に適用した結果、改善サイクル（CAPDo）の入り口として有用であることが

確認できた。

・PJ の課題が明確になり、改善策が立案可能

・改善の前後比較で改善効果を見える化

・顧客との認識相違が明確になり、コミュニケーションが活性化し

－顧客と合意すべき改善ポイントが明確化

－顧客満足度の向上効果

さらに 2016 年度分の診断データを分析した結果、社内ノウハウの流通により改善が促進される可能性が高いと思われ、新しい施策であるエンハンスメント（運用/保守）革新活動へと発展した。

<キーワード>

保守、運用、生産性向上、品質向上、セルフアセスメント、改善、見える化、簡易アセスメント、

エンハンスメント、社内ノウハウ

<想定する聴衆>

保守、運用の生産性向上、品質向上などの改善に携わっている方、関心をもたれている方

PJ マネージャ、PJ リーダ、改善担当、品質担当

＜活動時期＞

2014/10～現在　継続中

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（　　）

<発表内容>

1.背景

昨今のシステム開発は、売上・利益ともに、保守の比率が高くなっており、保守の生産性向上/品質向上が喫緊の課題となっている。また保守と並行で運用を行なっている組織もあり、運用へのプロセス改善のニーズも高い。

このため、保守/運用の課題を見える化するツールとして「あるある診断ツール」を作成し普及活動を行ってきた。

さらに本ツールの結果データを分析し、改善をし易くする仕組み造りを始めた。

2.改善したいこと

新規開発に関しては社内標準プロセスがあり、PPQA、工程開始/終了クライテリアなどの制度により生産性向上や品質向上の問題/課題への取組はできあがっている。しかしながら、保守/運用に関しては標準プロセスが未整備であり、日常業務で問題/課題への意識がしづらい状態となっている。

この様な状況のため、保守/運用 PJ の生産性や品質に関する問題/課題を見える化する仕組み、改善に繋げる仕組みが必要である。

3.改善策を導き出した経緯

「あるある診断ツール」は以下のコンセプトで開発した。

・誰でも簡単に
・時間をかけずに（約 30 分）
・生産性や品質の低下を招いている、良くある事象をチェックするだけで
・PJ の課題を、8 つの視点から点数化して見える化することができる
・自己診断（セルフアセスメント）のための簡易アセスメントツール

本ツールを適用し、その診断データ（98PJ（441 名、約 11.6 万件のデータ））を設問の平均点と標準偏差でゾーン分析した結果、

・最優先で改善すべき課題が見える化され
・課題の解決策が社内に存在する可能性が高い事が判明した（社内ノウハウの存在）

4.改善策の内容

本ツールはセルフアセスメントツールの一種である。誰でも簡単に回答できるようにするため、設問は生産性低下や品質低下を招く、良くある事象の有無をチェックする方式とした。

設問への回答は「有る」、「どちらかと言えば有る」、「どちらかと言えば無い」、「無い」の 4 択としている。回答結果は 8 角形のレーダーチャートで示され、問題が存在する箇所が視覚的に分かるようになっている。回答結果データはエクスポート/インポートすることが可能なため、他者や、他 PJ、他部署などのデータと比較することができ、改善ポイントの見える化だけでなく、様々な使い方ができるようになっている。

本ツールの診断データを分析した結果、課題の解決事例が社内に存在する可能性が高い事が判明し、事例のスムーズな流通が必要であることが分かった。スムーズな流通のためには土台となるプロセスや用語の標準が必須となるため、保守/運用プロセス標準を早急に整備する事となった。

5.改善策の実現方法

ツールの設問は、共通フレーム 2013 のアクティビティーや、SPEAK-IPA のプラクティス、運用の場合 ITIL のプロセス等を参考にしつつ、実施していなかった場合起こると思われる問題を想定し作成した。その文言は、なるべく短く、誰でもが理解できる用語を使うよう考慮した。

また、事例の流通に必要な保守/運用のプロセス標準に関しては、保守/運用を顧客へのサービス提供であると捉えてサービス

の標準プラクティスである ITIL を参考にして作成した。
さらにこの保守/運用標準プロセスは以下を考慮している。

・守りではなく攻めのサービスを提供する
・保守/運用をエンハンスメントと言う言葉に代えてポジティブな仕事として位置付け
・定常業務として継続的な改善プロセスを回す仕組み

6.改善による変化や効果

「あるある診断ツール」は、以下の通り、改善活動（CAPDo）の入り口、成果の確認として有効であることが認識され、現場の PJ に浸透しつつあり、現在 30%強の PJ で定期的な利用をしている。

・PJ の課題が明確になり、改善策が立案可能
・改善の前後比較で改善効果を見える化
・顧客との認識相違が明確になり、コミュニケーションが活性化し
  – 顧客と合意すべき改善ポイントが明確化
  – 顧客満足度の向上効果

しかしながらまだ多くの PJ は一度限りのイベントとしての利用に留まったままである。
その理由として、改善活動のメリットが部門経営層に届いていない（コストだけ使っていると思われている）、改善活動の時間が取れない（時間外活動だと思っている）といったものである。
このような状況を変えるために、保守/運用（エンハンスメント）プロセス標準の中に継続的改善活動を定め、定常業務の一環として改善を位置付けている。
今後、このプロセス標準を利用したエンハンスメント革新活動を推進し、改善が無理なく進み定着するように推進する。

7.改善活動の妥当性確認

上記で述べた通り、「あるある診断ツール」は改善活動（CAPDo）を回す取り掛かりのツールとしての位置付けは十分満足している。
エンハンス標準プロセスに関しては、活動が開始されたばかりであり、浸透にはまだ時間がかかりそうである。

A. 参考情報

[1] ・独立行政法人　情報処理推進機構　技術本部　ソフトウェア・エンジニアリング・センター編、
プロセス改善ナビゲーションガイド～自律改善編～、2013
[2]・独立行政法人　情報処理推進機構　技術本部　ソフトウェア・エンジニアリング・センター編、
共通フレーム 2013、2013
[3]・独立行政法人　情報処理推進機構　技術本部　ソフトウェア・エンジニアリング・センター、
SPEAK-IPA（Rev.1.0.2.0）

1A3「文書作成支援ツールによる組織開発力の強化」中村伸裕（住友電工情報システム)

<タイトル>

文書作成支援ツールによる組織開発力の強化

<サブタイトル>

<発表者>

氏名（ふりがな）：中村　伸裕（なかむら　のぶひろ）
所属：　住友電工情報システム株式会社

<共同執筆者>

氏名（ふりがな）：
所属：

<主張したい点>

住友電工ではシステム開発で作成する文書の作成・管理ツールを 2005 年に自社開発し、生産性の向上やノウハウの共有を図ってきた。文書は Web ベースで作成され、社員はブラウザで全システムの文書を閲覧することができる。今回の発表では、Web 化の課題、ツールの機能、効果について説明し、共有ディスクと表計算ソフトを中心とした文書作成・管理を行っている組織に対して改善のヒントを提供する。

<キーワード>

文書作成、文書共有、成果物ベースライン、改訂管理、生産性向上、ツール

<想定する聴衆>

改善推進者、開発者

＜活動時期＞

2005～2017 年　（現在も継続中）

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（　　）

<発表内容>

1.背景

2005 年当時、住友電工のシステム技術グループはシステム開発で使用するミドルウェア等の技術資料を HTML で作成し、ブラウザで閲覧できるようにしていた。技術の評価部隊であるため HTML にも精通しており、テキストエディタのみで技術資料のサイトを作成していた。しかし、時間が経過し、技術資料の改訂や引き継ぎが発生すると以下のような問題が顕在化した。

(a) 改訂の生産性が低い

作成された HTML ファイルには見栄えをよくするためのタグが多く含まれているため、安易に修正すると画面表示が乱れたりすることもあり、改訂作業の効率が低かった。

(b) 作成者、発行日等が不明

作成者、作成日、発行日等の文書情報が表示されておらず、古い情報かどうか確認できない。

(c) 履歴が残らない

変更時、HTML ファイルを直接編集しているため、過去の文書を閲覧することができない。また、いつ、どこが変更されたのか容易にわからない。

これらの問題を解決するため、文書作成・管理ツールを自社開発した。このツールはシステム開発でも利用できると判断し、横展開することにした。

本発表では、システム開発や改善活動に関わる文書作成・文書管理ツールについて、２章で開発時の要件、３章で得られた成果、４章でシステム構成、５章で成果を得るための仕組みについて述べる。続いて６章でツールの保守状況を示した後、７章で改善活動の妥当性を示す。

2.ツールの要件

全社のシステム・ドキュメントを Web ベースで統合管理するにあたり、以下のような要件を設定し、ツールの自社開発を進めた。

(a) 権限管理

ノウハウ共有のために社員はすべての文書が閲覧できる。ただし、文書の更新はシステム担当者のみに制限される。また、パートナー企業の開発者は担当するシステムのみ閲覧・更新できるようにする。

さらに、予算管理等の機密性の高い資料は管理者権限を持ったユーザーだけが閲覧・更新できるようにする。

(b) HTML 文書作成の効率化

システム開発では表形式で仕様を示すことが多いが、HTML で表を作成する場合、

<TABLE><TR><TD></TD></TR></TABLE>等のタグを多数記述する必要があり、作成効率が悪い。また、修正時もダグ構造の理解が必要で保守性が悪い。できるだけ HTML のタグを書かないで見栄えのよい文書を作成したい。

(c) 作成者等の文書情報の管理

本文とは別に、作成者、作成日、作成部署、発行者、発行日、発行部署等のメタデータの管理を行い、文書閲覧者がブラウザで確認できるようにする。

(d) 旧版文書の保管

文書が作成中のドラフト版か正式発行されたものなのかわかるようにする。改訂された場合、過去の文書も閲覧できるようにする。

(e) 管理資料の自動生成

文書の作成計画や実績工数の登録によりＥＶＭが自動出力でき、文書作成のコスト、スケジュールの状況が把握できるようにする。また、レビュー記録の登録により管理図が自動出力でき、品質の状況を把握できるようにする。

3.成果

２章で述べた要件を満たすツールを開発し、現場のニーズに応じて機能追加を続けた結果、以下の成果が得られている。

3.1 ツールによる成果

文書作成および文書管理をツールで支援するとで、以下の成果が得られている。

(a) 文書作成の効率化

外部仕様書、プログラム仕様書、議事録等、文書の種類毎にテンプレートを提供し、文書作成時に自動的にテンプレートをコピーすることで、文書作成が効率的にすすめられるようになった。

(b) 閲覧性の向上

文書は URL で簡単に指定できるため、メール等で簡単に連絡することができる。また、表計算ソフト等で作成した文書の閲覧はダウンロードやツールの起動に時間がかかるが、HTML 形式の文書は高速に閲覧できる。

(c) ソフトウェア品質の安定化

レビュー・テスト記録の登録により管理図が生成できるため、管理図の活用が定着している。その結果、システムの品質が安定し、プロジェクトが円滑に進むようになった。[1][2]

(d) 成果物ベースライン設定

CMMI の構成管理で要求されている成果物ベースラインの確立が効率的に行えるようになった。

(e) 会議の効率化

会議中にリアルタイムで目標施策関連図や因果関係を示すツリー図等を作成することにより、空中戦による意見のすれ違いを防ぐことができている。

(f) ペーパーレス会議

会議開催案内で資料の URL を連絡しており、プロジェクタで資料を投影して会議を進めるが、参加者は自分のＰＣで資料を確認することができる。紙資料の配付はほとんど行っていない。

(g) 最新技術の活用

機械学習等の技術が簡単に使えるにようなった。

3.2 文書共有による成果

ツールを開発し、組織全体で作成した文書が共有できるようになった結果、以下の成果が得られている。

(a) SQA 検証の効率化

各プロジェクトの資料はＰＭとコンタクトしなくても確認できるため、月１回実施しているＳＱＡが効率的に進められる。例えば、管理図による品質の監視を行っているかどうか、EVM でコストとスケジュールの確認を行っているかは、これらの文書を直接確認すればよい。

(b) プロセス実績ベースライン作成の効率化

欠陥データがツールで管理されているため、基準値やベースラインを作成する際、効率的にデータを収集できる。ベースライン設定時のプロジェクト側の作業は異常値の確認程度の軽い負荷ですんでいる。

(c) 各プロジェクトの効率化

他プロジェクトの文書をサンプルとして利用することで初心者はスムーズに作業に着手できるようになった。

また、連携するシステムの仕様が確認できるため、連携方法の設計が進めやすくなっている。

(d) 派生開発の効率化

派生元から派生先に最新版の文書が簡単にコピーでき、効率的に文書が作成できる。

(e) 改善活動の効率化

全システムの文書が改善活動の INPUT として利用できるため、現状把握がしやすい。

4. システム構成

4.1 サーバー構成

図 1 に文書を保存するサーバーの構成を示す。2005 年当時、拠点間のネットワーク速度が遅かったため、開発拠点毎にサーバーを設置し、レスポンスを確保した。SW 生産管理と書かれたサーバーは共用開発サーバーに開発環境を作成するシステムで以前から稼働していた。このサーバーにはシステム毎の担当者が記録されている。この情報を活用して、各地区に設置されている文書サーバー内のシステム毎の共有ディスクに権限設定を行う。システム担当者には共有ディスク(samba)の権限を設定し、文書を閲覧するための Web サーバーの権限は社員と協力会社の担当者に付与する。類似情報検索のサーバーはすべての文書サーバーの文書を収集し、文書中のキーワード検索ができるシステムである。LDAP は社内業務システムのユーザーID、パスワードと連携し、本ツールでユーザー認証を行うために利用される。

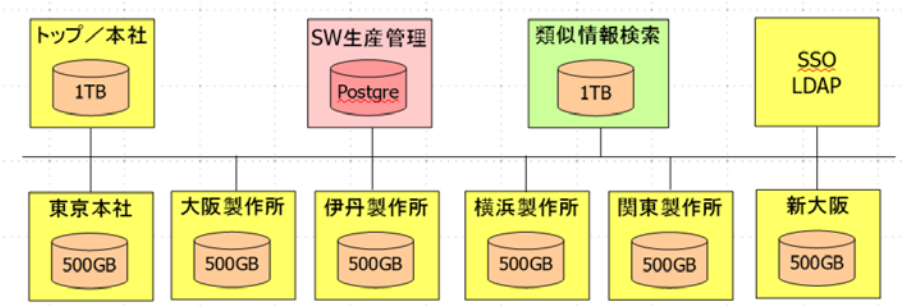


図 1． 文書管理環境の構成図

5. システム機能

ここでは 3 章で示した効果を得るための仕組みを説明する。

5.1 文書作成の基本動作

(1) 基本動作

図 2 にツールの動作イメージを示す。ＰＣには自社開発したツール、テキストエディタ、ブラウザがインストールされている。サーバーでは Apache Web Server および 共有フォルダ機能を提供する Samba が稼働している。文書作成者がツールを使って新規文書作成ボタンを押すと、サーバー上に本文ファイルが生成され、テキストエディタが起動する。同時に作成者、作成日等のメタ情報が記録される。作成者は図 3 のような文書を編集し、Web への反映ボタンを押すとサーバー上に index.html ファイルが生成され、ブラウザで図 4 のような作成者等のメタ情報が追加された文書が閲覧できる。HTML 文書の作成工数を減らすための工夫が %% で始まるコマンドである。図 3 では、%%table コマンドはテーブル形式でデータを表示するコマンドである。”<tr><td>” といったダグを書かずに CSV 形式のデータでテーブルが表示できる。この機能により、文書の記述量が少なくなり、文書改訂時も直感的に修正できるようになった。また、%%attach コマンドは所定のフォルダに添付されているファイルのリンクを表示するコマンドである。当時は日本語名称のファイルのリンクが簡単にできなかったため、手軽に日本語の添付ファイルが扱えるようになり、閲覧性が向上した。

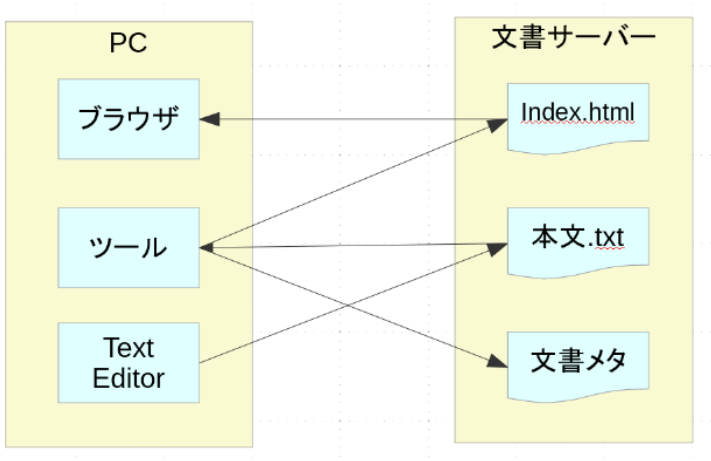


図２． ツールの動作イメージ

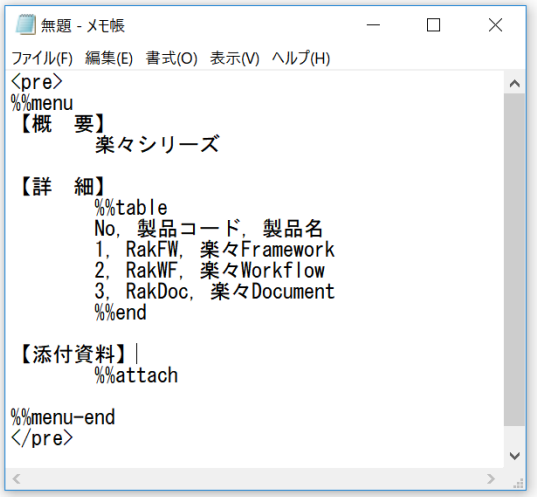


図３． 作成者が編集する本文

図４． ブラウザに表示される文書例

(2) コマンド

本ツールで利用できるコマンドはプロセス改善と同期して拡張しており、現在、約 100 種類用意されている。主な機能を表 1 に示す。

表 1． ツールの主なコマンド

機 能	説 明
画像	画像を文書中に埋め込む。拡大縮小や注目箇所に赤丸や四角をオーバーレイできる。
業務フロー	業務レーン毎のタスクを HTML で表示する
USDM	要求を清水吉男氏が提唱されている USDM 形式で出力
ツリー	ツリー構造を HTML で出力する。なぜなぜ分析等に利用。
グラフ	折れ線グラフ、棒グラフ、円グラフ、相関図、箱ひげ図、ＥＶＭ、管理図等を SVG(Super Vector Graphic)を使って出力する
GQM	目標施策関連図（図５）、GQIM Tree 等を SVG で出力
信頼度成長曲線	CSV 形式のテストデータから信頼性成長曲線を SVG で出力。アニメーションで各時点の成長曲線も確認できる。[3]
ToDo	文書中に後で修正が必要な箇所にメモが書ける。文書を発行する前に残作業がないか確認できる

本ツールの入力はテキストファイルであるが、図 5 に示すような図を自動レイアウトする機能があり、作成効率が向上する。

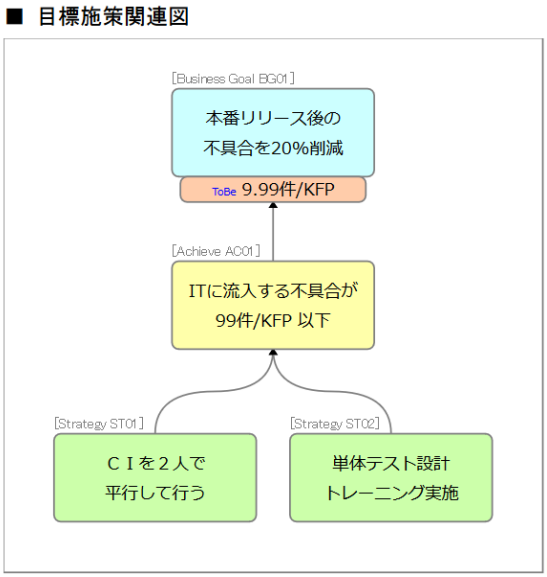


図 5． 目標施策関連図の出力例

5.2 改訂履歴の管理

(1) 改訂管理

新旧の文書がいつでも閲覧できるよう、各文書の URL を以下のように定めた。

http://サーバー名/文書種類/システムコード/サブシステムコード/フォルダ ID/文書 ID/Rev.999/index.html

Rev.999 の部分が版を示している。URL の文書種類は、標準文書、システム文書、WG 文書の種類を示す記号である。一般的な閲覧方法は文書一覧を表示後、各文書に遷移する手順であるが、文書一覧からは最新版の文書に遷移するようにしているので版を意識する必要はない。

(2) トレーサビリティ

上記機能は、Rev.001，Rev.002，… と一つの文書の履歴を管理するものであるが、さらに前後で作成される文書の関係を記録する機能を追加した。ワープロソフトで作成した文書は、数百ページに及ぶことも多く、成果物間の関係を記録することが難しい。しかし、外部仕様書を HTML で作成すると受注画面、出荷画面といった機能が 1 つの文書として作成されるため、プログラム仕様書、ソースコードの前後関係を明確にすることが容易になった。図 6 に文書の改訂が繰り返された際の例を示す。プログラム仕様書は外部仕様書の Rev.002 をベースに作成されたが、後に外部仕様書が改訂されたため、Rev.003 の外部仕様書に対応したプログラム仕様書 Rev.003 を作成している。このような管理機能を実装することで先行文書が改訂された後、後続文書が改訂されていな状況が簡単にわかるようになり、変更の実績管理が簡単にできるようになった。

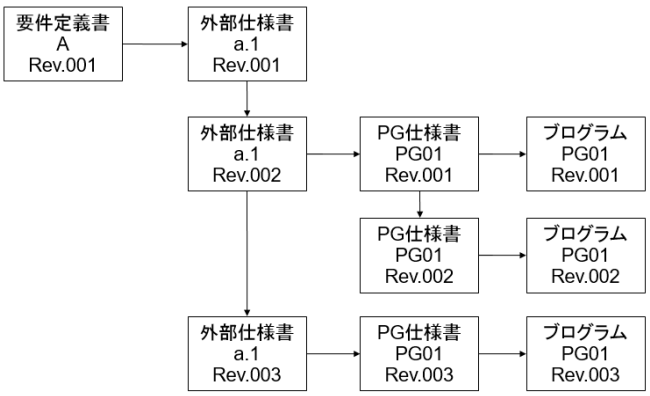


図 6． 改訂履歴と前後の文書の関係

(3) 派生開発の支援

最近では自社開発パッケージ等、システム全体を再利用する活動が増えており、派生開発の機能を追加した。ベースとなるソフトの文書（最新版）をこれから開発するシステムの文書環境にコピーでき、派生開発環境が効率的に作成できる。また、開発後に、元のソフトに取り込む機能があった場合、元の文書環境に最新版としてコピーすることができる。ソフトウェア・プロダクト・ラインの観点で有用な機能だと考えている。

5.3 成果物ベースラインの設定

ソースコードのベースラインの管理は SubVersion や Git の管理機能を利用して行っている組織が多いが、文書のベースラインの設定は表計算ソフトを使ってベースラインに含めるファイルを手作業で行っている組織が多いのではないかと考えている。当組織では CMMI のアプレイザルを行うまで成果物ベースラインの設定はあまり行われていなかった。そこで当ツールの履歴管理機能を強化して簡単に成果物ベースラインが設定できるようにした。図 7 に成果物ベースライン設定の仕組みを示す。本ツールで作成した文書はドラフト版になっている。レビュー記録が登録されるとその版の文書は凍結され、新しい版として文書を修正する。修正が終われば、発行という操作で Ver.番号が付与される。図 7 では a.1 という文書が発行後に変更要求があり、修正作業が行われている状況を示している。点線で示される時点でツールのリリース機能を動作させるとその時点で発行されている版の一覧を作成し、その時点の文書一覧がブラウザで閲覧できるようになる。この一覧が構成監査を経て成果物ベースラインとなる。開発者は表計算ソフトにファイル名を記録する必要がなくなり、また、ベースラインに含まれる文書はリンクをクリックするだけで閲覧でき、閲覧性が高い。さらに 2 回目以降のベースラインを作成する際、前回ベースラインとの差分を、各文書の改訂記録をまとめて自動生成している。システムの開発規模が膨らんだ際、顧客との費用負担の交渉で活用されているケースもある。

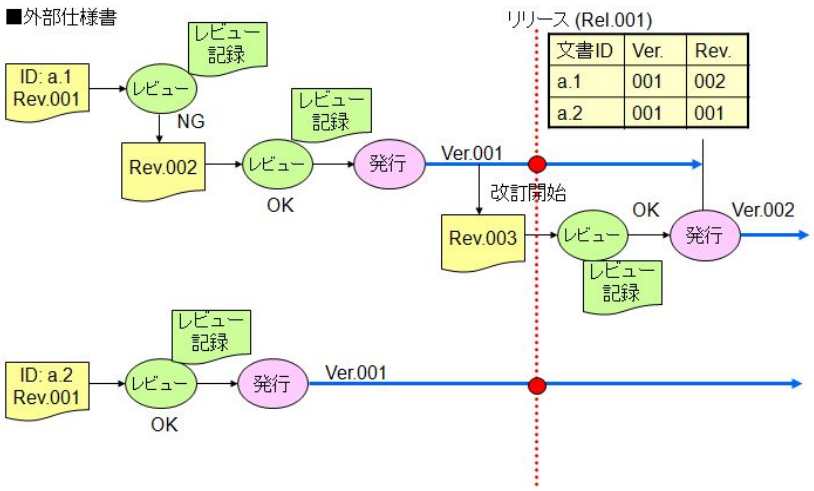


図 7． 成果物ベースライン設定の仕組み

5.4 品質管理機能

CMMI Level 5 達成を目指して改善活動を進めた際、管理図の出力機能を追加した。構成を図 8 に示す。上部の欠陥管理機能が追加された部分である。レビュー者は文書をレビューし、欠陥の内容を登録する。ツールは、レビュー対象成果物の規模を自動測定し、レビュー記録の欠陥数をカウントする。品質集計部門は毎年、基準値を登録しており、これらの情報から図 9 に示すような管理図が生成される。管理図の機能を普段使用しているツールに組み込んだため、管理図を使った品質管理がスムーズに社内展開できた。

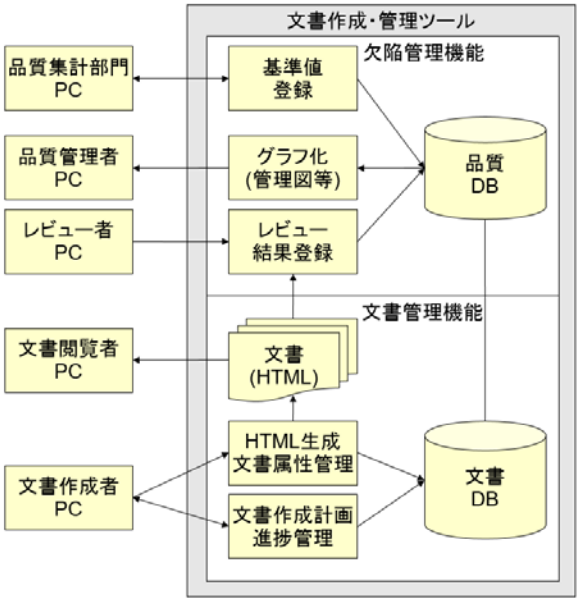


図８．欠陥管理機能の構造

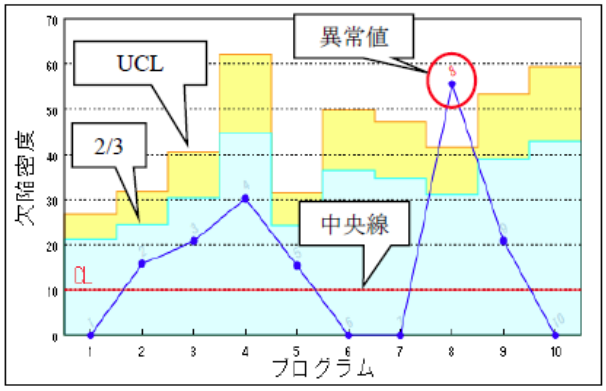


図９．欠陥密度の管理図の例

5.5 その他の機能

(1) リアルタイム更新機能

本ツールはテキスト形式の本文ファイルから図・表を生成できる。この機能は会議中に本文ファイルを編集しながら、プロジェクタの画面に図を表示するといった使い方もできる。例えば、課題、施策の議論をしている最中に目標施策関連図（図５）を映し出すこともでき、ありがちな空中戦による認識違いが発生せず、効率的に議論を進めることができる。業務フロー図やなぜなぜ分析等も同様の方法を使って効率的に進めることができる。

(2) 機械学習のサポート

最近のはやりである機械学習ツール(Apache Spark)と連携し、データの分類、予測ができるようにした。ツールは機械学習のソースコードを自動生成し、指定されたファイルのデータを学習し、結果を HTML で出力できる。この機能により、クラスタリング等データ分析が手軽に実施できるようになった。

6. ツールの保守

システム開発で利用するツールはあれば便利というレベルではなく、ないと仕事ができないレベルに浸透することが望ましい。そのため、ツールの強化はニーズが出てくるたびにタイムリーに対応している。図 10 は 2005～2017 年のツールの物理行数の推移を示しており、継続的に機能拡張されていることがわかる。2016 年には信頼度成長曲線や GQM のコマンドが追加され、現在、ツールのソースコードは約 150KStep になっている。

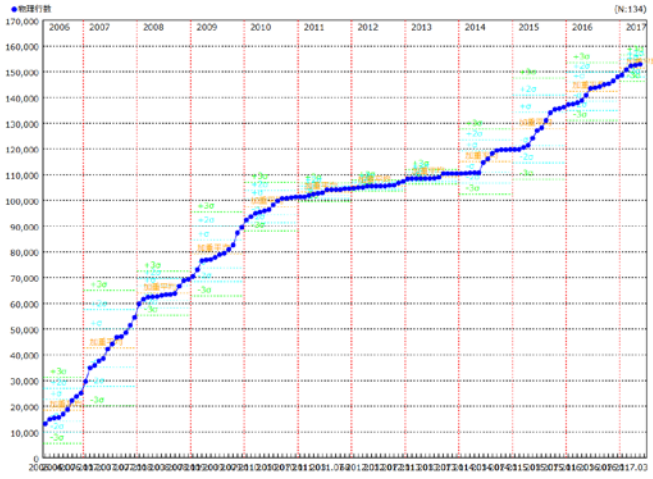


図 10. ツールのライン数の推移

7.改善活動の妥当性確認

2017 年 5 月現在、本ツールに登録されている文書は、826,016 件であった。件数には改訂された文書を含んでおらず、毎日、250 以上の文書が新規作成されている計算になる。システムに関係する文書は 761,059 件で 92%を占めており、多くのシステム開発で利用されていることがわかる。システム以外の文書として、開発標準、技術資料、ＷＧ活動の記録等が格納されている。

本ツールは技術資料の管理上の問題点を解決するための簡単な機能から始まり、現場のニーズに応じて連続的に機能拡張を行ってきた。個々の機能改善では改善前と改善後の差が小さく、定量的な評価はできていない。

一方、当ツールを使用している開発部門は 2011 年から CMMI Level 5 を継続しており、リード・アプレイザーからは「当組織の最大の強みは文書作成・管理ツールである」という評価を得ている。CMMI の CM、MA、PPQA、PMC、IPM、QPM、OPP、OPM 等のプロセスエリアで強みになっている。

8.まとめ

今回紹介したツールは 10 年以上の期間をかけて改善を続けてきたもので、すぐに他の組織で展開できるものではない。しかしながら、システム開発の 2 大成果物は、文書とプログラムであり、文書の作成・管理を改善する価値は高いと考えている。メーカーでは製造設備をそのまま使うのではなく、改良して生産性を高めている企業が競争力を持っている。ＩＴ業界でもこのような取り組みが必要ではないか。

A. 参考情報

[1] 中村伸裕, “統計的品質管理手法の確立”, SPI Japan 2008, 2008
[2] 山邊人美, “統計的品質管理手法の全社展開”, SPI Japan 2008, 2008
[3] 中村伸裕, “信頼度成長曲線の導入による統合テストの改善”, SPI Japan 2016, 2016

1A4「ユーザワーキングをプロセスに組み込んだ品質向上」山中美圭代（ダイキン情報システム）

<タイトル>

ユーザワーキングをプロセスに組み込んだ品質向上

<サブタイトル>

ユーザと共に創る満足度の高い仕様

<発表者>

氏名（ふりがな）：山中 美圭代（やまなか みかよ）
所属： ダイキン情報システム株式会社 開発５部

<共同執筆者>

氏名（ふりがな）： 宮脇 隆志（みやわき たかし）
所属： ダイキン情報システム株式会社 開発５部

<主張したい点>

品質向上には不具合が発生しない完成度の高いシステム作りと共にユーザ満足度の高い仕様の実現が不可欠である。特に後者への取組みは投資効果が得られる費用のなかで全てのユーザが満足するものを実現するのは難しい。IT 部門で開発する我々が、設計者（ユーザ）が実施する作業内容や設計全体の方向性を考える管理部門の意向（開発依頼元）を十分理解して両者の満足度の高い仕様を提案するのは困難であった。今回の当社の取組みとしては開発着手前に実際のユーザが実施する作業内容を経験し、管理部門の意向も理解することで、両者の満足度を満たす仕様を提案できた工夫事例を報告する。

<キーワード>

プロセス改善、ユーザ満足度の高い仕様、ユーザワーキング、管理部門、ユーザ

<想定する聴衆>

ＳＥＰＧ、PM

＜活動時期＞

2016/４～現在

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（ ）

<発表内容>

1.背景

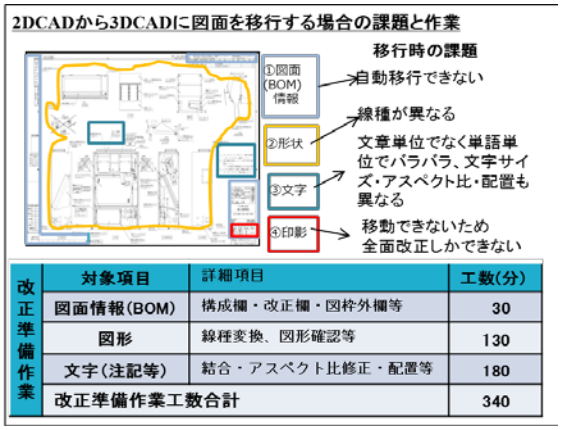
私の所属する開発 5 部は当社設計部門の開発企画グループからの依頼を受け設計にかかわる社内システムの開発を手掛けることが多い。開発企画グループは設計部門の管理統括を行い、海外拠点を含む全社での設計業務の統一化・効率化を推進する管理部門である。 全体の方向性を重視する開発企画グループから提示される仕様要求はシステムユーザである設計者の要望を十分に満たす要求とならないことが多く、その結果構築したシステムがユーザになかなか浸透しないケースも発生していた。今回のプロジェクトでは下記の課題があり、CAD 統一化を進めるためには、どうしてもユーザ要望を満たすシステムを早期且つ投資効果範疇の費用で構築する必要があった。 そこでユーザ作業を実体験した上で、作業経験を生かし、作業可能な最低限のプロトタイプを作成した。そして、そのプロトタイプを使い開発企画グループと共にユーザワーキング（ユーザ試行とヒアリング）を実施し、段階的にユーザ満足度の高い仕様を実現してきた内容を報告する

【課題】

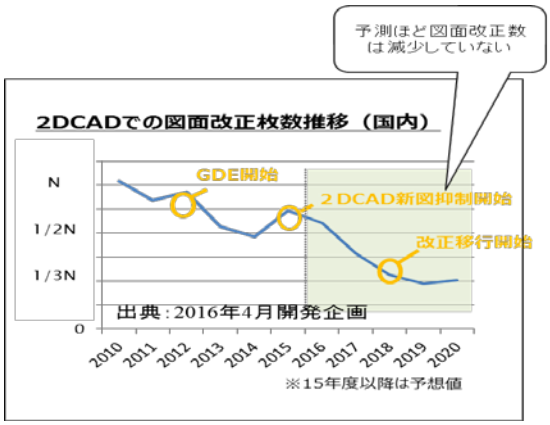
1）海外拠点を含む全社で CAD を統一化を実施しているが、それには古い 2 DCAD で改正している図面（年 1 万枚程度）を 3 DCAD へ移行する必要があるが、手作業での移行は多大な工数負荷が発生するため移行できていない。

2） 2 DCAD と 3 DCAD のソフト間互換性がないため通常変換では精度が悪い

3）システム化してもユーザ（設計者）が使ってもいいと思える仕様でないと使用せず、移行が進まない



改正準備作業	対象項目	詳細項目	工数(分)
	図面情報(BOM)	構成欄・改正欄・図枠外欄等	30
	図形	線種変換、図形確認等	130
	文字(注記等)	結合・アスペクト比修正・配置等	180
	改正準備作業工数合計		340



2.改善したかったこと

1）自動変換率のよいシステムの構築

2）変換精度の高いシステムの構築

2）ユーザが求めているシステム要求を仕様に取り入れることのできるプロセスに改善する。

3.改善策を導き出した経緯

開発に着手する 1 年前に手作業で 2 DCAD から 3 DCAD へ変換する作業（年間 100 枚以上）を請負った。 その作業の中で、ユーザがどこで困っているのか、何に時間がかかるのかを実感し、ユーザが望む仕様の全貌を理解できたことから今回の改善が導けたと思う。また、システム化を念頭に置いていたため、並行して下記の調査・検討も実施した。

1）自動変換率を良くするには図面情報をはじめ手作業で再入力する部分を自動化できないか調査した。

2）変換精度を上げるのに印刷は正しく出力されることから印刷に使用しているファイル（HPGL）を使用できないかと調査した。

3）早期システム化が必要であったため、開発当初からシステムをステップ 1 、2 と分けることが決まっていたことからステップごとに仕様変更及びユーザの要望を何らかの形で反映できないか検討した。

4.改善策の内容

1）自動変換率のよいシステムの構築

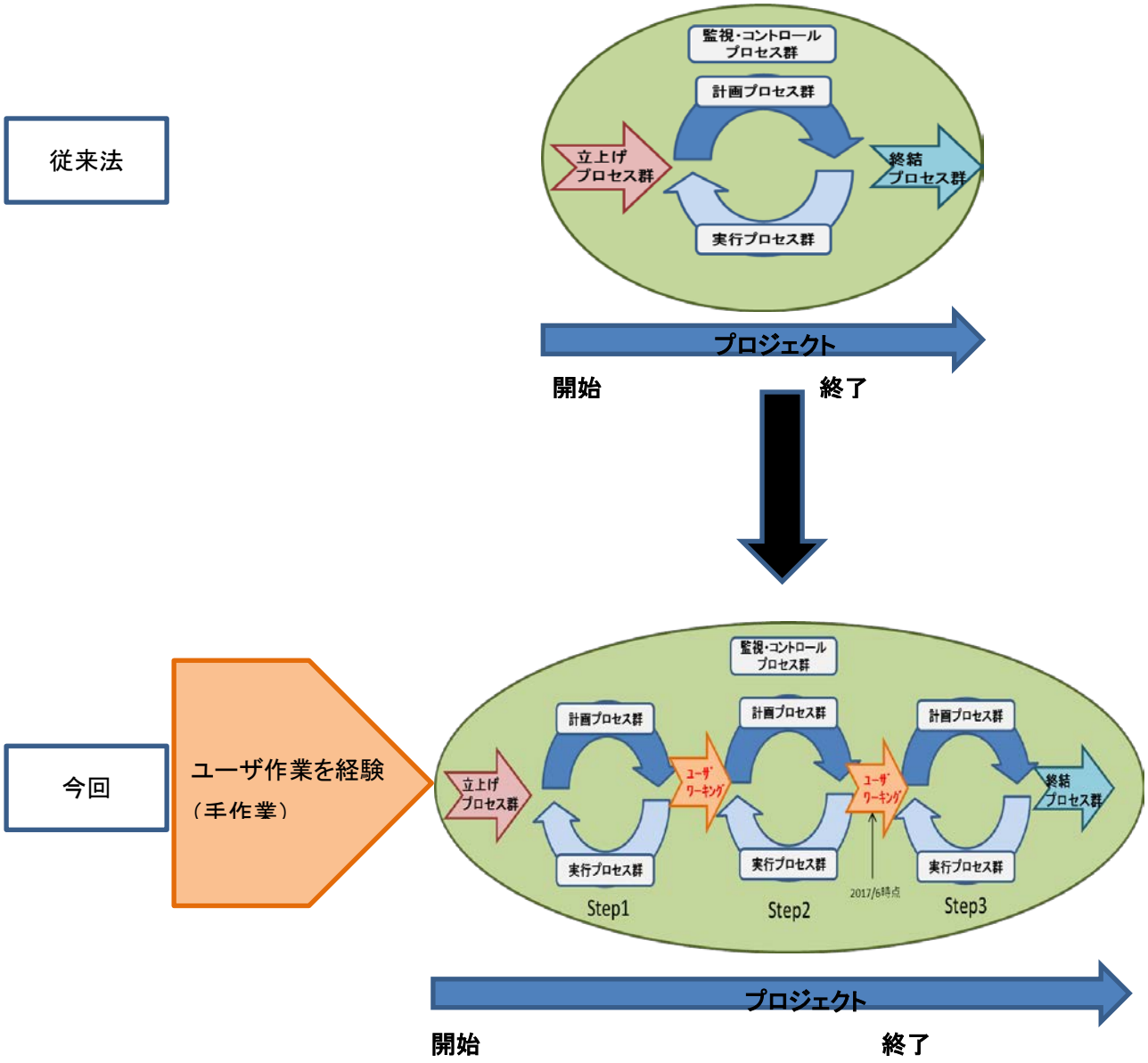
図面情報（BOM）を２DCAD 図面から出力し、３DCAD でその情報を自動で読み込む仕組みを作成した。

2）変換精度の高いシステムの構築

HPGL 変換を利用することにより、変換精度は向上するが、修正する場合の使い勝手（図形編集等）はあきらめる等優先すべき項目と目的（改正のため修正は少ない）に沿ったシステム構築を実施した。

3）プロセス改善

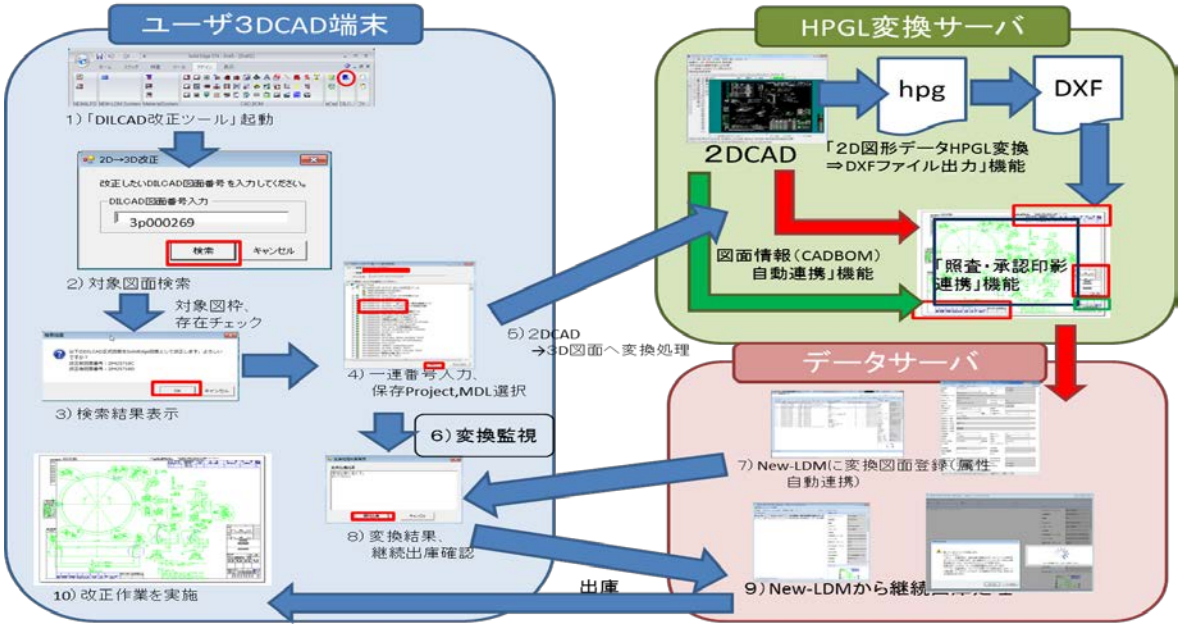
下記のようにプロセスを変更し、事前経験期間の追加とプロセス自体を何回も繰り返し、そのステップ間にユーザワーキングを入れることで早期システム化とユーザ要望を反映するようにした。



5.改善策の実現方法

1）自動変換率がよく変換精度の高いシステムの構築

具体的には下記のようなシステム構成とし、ユーザはローカル端末で保存場所と変換図面番号を入力するのみで２Ｄから３Ｄ図面へ変換することを可能にした。



2）プロセス改善

ステップ間にユーザワーキングとしてフィールドテスト（一部ユーザに一定期間使用してもらう）を実施してもらい、その結果よりユーザ要望・不具合を会議形式でヒアリングする機会を開発依頼元（開発企画）主体で実施することを盛り込むことにした。そこで得られたユーザ要望を開発依頼元と討議し、優先順位をつけて次のステップで盛り込むというプロセスにすることで、ユーザ要望をシステムに取り込んでいった。

6.改善による変化や効果

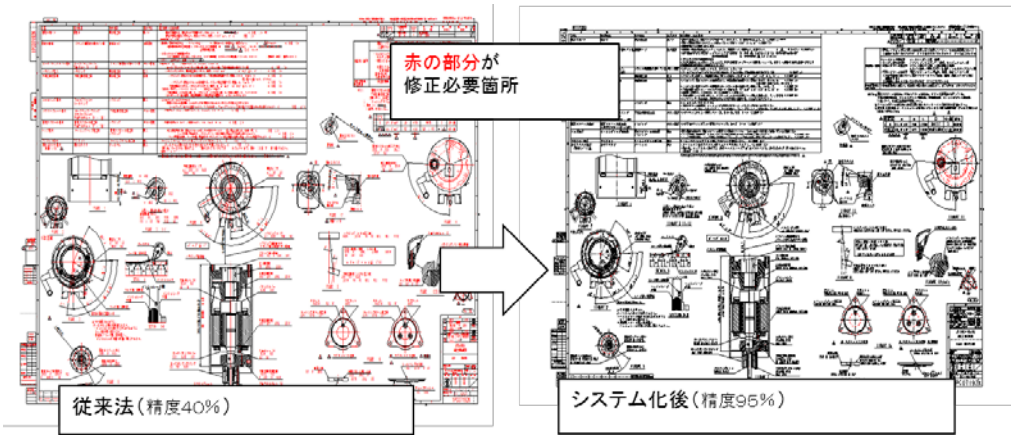
1）自動変換率の改善

従来法に比較して**95%以上の自動化**が実現でき、ユーザが２Ｄから３Ｄ化する**工数負荷を軽減**(340 分→10 分)できた。

変換対象	従来（DXF直接）	HPGL変換 ステップ1	HPGL変換 ステップ2
図面情報	手動　30分	手動　5分	自動　0分
図形	手動　70分	自動　0分	自動　0分
	目視確認　60分	目視確認10分	目視確認10分
文字（注記等）	手動修正180分	不要（修正不可）	不要（修正可）
合計時間	340分	15分	10分
自動化	0%	95.6%	97.1%

２）変換精度の高いシステムの構築

手動では **４０%の変換精度を９５%に向上できた**



３）ユーザが求めているシステム要求を仕様に取り入れることのできるプロセスに改善

・ユーザにプロジェクトに対する評価アンケートを実施した結果は **0～４の５段階で 3.4 と高評価**

・ステップ３までは一部ユーザにのみ説明会実施し、限定使用としているが、コマンドは全ユーザ端末にインストールしていたところ口コミで広まり、使用者が拡大している。

7.改善活動の妥当性確認

６－３）の結果より改善活動は妥当であったと考える。

1B1「医療現場の IT 製品開発でスクラム・オブ・スクラム　-組織のマインドを変革する！-」清水弘毅（オリンパス）

<タイトル>

医療現場の IT 製品開発でスクラム・オブ・スクラム　-組織のマインドを変革する！-

<サブタイトル>

<発表者>

氏名（ふりがな）：清水弘毅
所属：　オリンパス株式会社

<共同執筆者>

氏名（ふりがな）：
所属：

<主張したい点>

アジャイルとは「常に改善し続ける状態であること」。スクラムを活用し、職場の現場メンバのマインド・生産性を改善し続けられる土台を気づいた。「言われたものをそのまま作る」のではなくメンバ一人一人がユーザの価値、品質を意識して開発に取り組む。そのためには、楽しく、共感でき、全員で考えられる開発体制が必要だった。開発に関わるメンバは約 50 人。４チーム・中規模のスクラムを、医療現場で利用される IT 製品という、品質が求められるシステム開発で実践。その取組内容を共有する。

<キーワード>

- １プロダクト複数チームのスクラム開発（スクラム・オブ・スクラム）
- 開発の楽しさを取り戻す
- 主体性を持つ

<想定する聴衆>

- ソフトウェアエンジニア
- スクラムマスター
- プロダクトオーナー
- プロジェクトマネージャー
- 品質保証の担当者
- アジャイル開発を実践してみたいと考えている方
- 職場を活性化させたいと考える方
- スクラム開発を複数チームで実践したいと考えている方
- アジャイル開発を小さく初めて、中規模レベルまでスケールさせたいと考えている方

＜活動時期＞

2016/02/01-継続中

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）

■ 改善活動を実施したが、結果はまだ明確ではない段階
□ 改善活動の結果が明確になっている段階
□ その他（　　）

<発表内容>

1. 背景

企業が大きくなれば大きくなるほど、組織は分断され、至るところに「壁」が生まれる傾向がある。オリンパスとしても例外ではない。弊社では、営業・開発・導入/保守・品質保証等様々な組織部門がある。開発内も、製品主査とソフトウェア開発に分かれており、さらにソフトウェア開発内では、開発部隊と試験部隊が分かれていた。ソフトウェア開発は、小会社であるオリンパスソフトウェアテクノロジー（以下、O-Soft）が担っていた。O-Soft として、プログラムを作る立場から見ると、様々なステークホルダーがおり、エンドユーザが誰なのかわかりづらい開発現場で、言われたものを作るだけだった。なぜ作るのかがメンバ全員に浸透しない。これでは仕事をしていても楽しさ、充実感が得られにくい。結果として、人が育ちにくい環境となっていた。

プロセスはウォーターフォールを採用していた。受け取った仕様からプログラムを作り、試験する。仕様変更があった場合は「変更管理プロセス」を回していた。変更管理のためにも非常に重厚なドキュメントを作成・周知のための会議に時間をかけていた。親会社であるオリンパスに設計書や試験仕様書のレビューをしてもらうためのレビュー計画が非常に綿密であったこともあり、「変更は良くないこと」のような雰囲気が漂っていた。

他部門に作りかけのシステムを見せるのに非常に敷居が高かった。ウォーターフォールプロセスであったため途中段階で見せにくく、また、開発途中で横槍が入るのを極端に嫌う文化があった。

上記のような背景のため、メンバはマインドが萎縮してしまい、主体的に動くことができなくなっていた。

2.改善したいこと

以下の点を改善したいと考えた。

- 組織の壁をなんらかの方法で壊したい。オリンパスも O-Soft も一体となって楽しんで開発できる体制にしたい。
- 開発を工程で分けず、開発者・ステークホルダー全員がエンドユーザのことを考えるようにしたい。「なぜ作るのか」をメンバ全員に腹落ちさせたい。
- エンドユーザのことを考えるのであれば、常に変化を受け入れたい（仕様変更等）。仕様変更のために無駄な会議を開き、時間を使いたくない。
- 定期的にシステムをリリースし続け、エンドユーザの満足度を高め、生産性向上に役立ちたい。
- 納期とコストは固定にし、要求を捉え直す、要件を考え直す。それが普通にできる状態にしたい。
- 医療現場で利用される IT 製品なので、より少ない労力で品質は確保し続けたい。
- 小さな失敗を繰り返し、大きな成功に繋げる、メンバたちをそんなマインドにしたい。
- 常に改善し続ける状態を目指す。

3.改善策を導き出した経緯

組織の壁を壊すには、組織をまたがった頻繁なコミュニケーションが必要であると考えた。この手段として、実際に作っているシステムを定期的に見てもらい、フィードバックをもらう。

リーダーやプロジェクトマネージャーが全ての仕様を掌握するのではなく、メンバ全員がなぜ作るのかを納得する。

上記を実現できるフレームワークである、アジャイル開発の一手法、「スクラム」を活用することとなった。

4.改善策の内容

偶然にも、ウォーターフォールで作ったシステムの一部機能が非常に使いづらいとのことで、作り直しのプロジェクトがあり、そこでスクラムを試したところ、開発者・ステークホルダー含め、非常に好評であった。特に、スプリントレビューが定期的に必ずあり作ったものに素早いフィードバックをかけられる点が評価された。

次に、スクラムでどうやって品質を確保するかという課題があり、品質保証部門にもチームに加わってもらい、スクラムを進めていった。結果、数値的にも成果が出始めた。このやり方でも製品開発できるのではないか、と自信が生まれた。

現在、上記の流れを引き継ぎ、製品開発 50 名体制で 1 プロダクト・4 チームスクラムを導入/保守も体制上ジョインしてもらい、

推進している。

5.改善策の実現方法

- ボトムアップ
  (ア) チームビルディングにて、チームの一体感を醸成する
  (イ) プロジェクトの思い出つくり
  (ウ) スクラムの愚直な実践
    ① スクラムをスクラムガイド通りやってみる
    ② 役割をあてる(Product Owner, Team, Scrum Master)
    ③ プロダクトバックログを作る
    ④ スプリント・プランニング～リファインメントまでの一連の流れの確実な実施
  (エ) スクラムマスターがスクラムを主導する。最初はとにかく引っ張ったり支援したり
  (オ) 技術的ボトルネックを把握するためにバリューストリームマッピングの実践
  (カ) XP の要素の実践（ペアプログラミング等）
  (キ) 社内外発表で成果をアピール・共有
  (ク) 品質特性によるチェックリストをスプリントプランニング時にストーリー毎に確認
  (ケ) アーキテクチャ・品質を常に意識する
  (コ) 堅牢化スプリントにより試験に集中するスプリントを作る
- 外部からの刺激
  (ア) 永和システムマネジメント様にコンサルティングをお願いする。（週 1～2 回）
    ① アジャイル相談会にて変化に対する心理的不安を軽減する
    ② 他社見学にてマインドチェンジを促す
    ③ プロダクトオーナー研修にてビジネス寄りのメンバにもスクラムを理解してもらう
- トップダウン
  (ア) 失敗奨励
  (イ) マネジメントと現場を繋ぐ場を定期的に設ける
- カンバン等による見える化の実践
  (ア) アナログとデジタルを併用して使う。どちらかというとアナログ重視。「やってる感」が周りに伝わる
  (イ) ユーザーストーリーマッピングやインセプションデッキ等、アジャイル手法をとにかく試す
  (ウ) 物理的な壁を作る
- 1 プロダクト複数チーム（スクラム・オブ・スクラム）
  (ア) スプリントのスケジューリング
  (イ) チーム毎に PO(Product Owner)の分身である APO(Area Product Owner)を作る

6.改善による変化や効果

この改善により以下のような変化・効果をもたらした。

- 各メンバが仕事を自分事として強く捉える用になった。開発を工程でわけず、要件定義～試験までを各チームで担当し、APO もチームに加わるため、ドキュメントベースでの無駄なやり取りはなくなった。
- これまでは特定の人に依存することが多かったが、チームで開発を進めるため、チーム内であればスキルが平準化された。
- メンバ全員が品質特性を強く意識して開発を着手することができており、機能・非機能の面で品質早い段階から作り込むことが出来ている。
- 我々のドメインでは、エンドユーザに直接話を聞くのは難しいため、その代弁者に近いステークホルダー（営業やインストラクタ

ー）と常にコミュニケーションを取りながら仕事を進めるようになった。その結果、開発の透明性が増した。

- 結果、会社間や部署間の壁を壊すことができた。

※余談であるが、このプロジェクトの途中で O-Soft は解散し、オリンパスに吸収合併された。親会社・小会社の壁はこれで完全に取り除かれた。

7.改善活動の妥当性確認

現在このプロジェクトは途中段階であるが、トライアル時点（2016/10-2017/3）での効果は以下の通り。

定量評価：

- 前システムの構築時と比べ、生産性が 20%向上。（生産性はコストをポイントで算出したもので表現）

定性評価：

- メンバからのヒアリング・アンケートの結果、本手法を継続したいという声が非常に強かった。開発現場に楽しさが生まれた。またステークホルダーからも取り組みについて好評価を頂いている。

A. 参考情報

[1] 社外発表 XP 祭り 2016 http://xpjug.com/xp2016-session-c7-1/

[2] O-Soft 統合 https://www.olympus.co.jp/ir/data/announcement/pdf/td161222.pdf

1B2「Agile 開発フレームワークを応用した効率的な開発ナレッジの獲得」林健吾（デンソー）

<タイトル>

Agile 開発フレームワークを応用した効率的な開発ナレッジの獲得

<サブタイトル>

自動車ソフトウェアにおけるプロダクトライン開発への適用事例

<発表者>

氏名（ふりがな）：林健吾（はやしけんご）
所属： 株式会社デンソー

<共同執筆者>

氏名（ふりがな）： なし
所属： －

<主張したい点>

ナレッジの少ないドメインの開発に携わるときに効率的にナレッジを獲得したい。Agile の開発フレームワークを応用して、何らかの実体を構築しながら小さく反復的に学習する方法を試みた。自動車ソフトウェアのプロダクトライン開発に適用し、効率的にナレッジ獲得できることが確認できた。

<キーワード>

ナレッジ獲得、Agile 開発、プロダクトライン開発、SPLE、自動車ソフトウェア、派生開発

<想定する聴衆>

ソフトウェアエンジニア、プロダクトライン開発者、組込系技術者、SEPG

＜活動時期＞

2016 年 12 月～2017 年 5 月（以降、継続中）

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動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、自動車システムの走行支援製品のソフトウェアを開発している。この分野は、技術競争が活発で市場の拡大も著しい。我々の組織ではプロダクトライン開発に取り組んでいる。その上で、２つの組織で技術競争と市場拡大に対応する組織体制を構築している。コア製品開発組織（以降、コア組織）は、技術競争に対応すべくコア資産を進化させながらコア製品を開発する。派生製品開発組織（以降、派生組織）は、市場拡大に対応すべくコア資産から派生製品を数多くリリースする。主体は派生組織であり、発表者は主体のマネージャを担う。本発表は、コア資産の変化の速さに、主体が対応しきれなくなることに危機感を覚え、活動を開始した。

2.改善したいこと

プロダクトライン開発では、コア資産を構築して初めの数製品をリリースした後は、資産の小変更とコンフィグレーションレベルの対応で派生製品を多数・長期間リリースして投資対効果を高めたい。しかし、我々の開発では技術競争が活発であるため、コア資産に追加される機能とアーキテクチャの改造が大きく、かつ毎年刷新され、コア資産そのものが毎年別資産化する状況が生まれた。その結果、派生組織では、コア資産の更新に対する知識・経験（以降、ナレッジ）を十分に補う時間を確保できない状態で、そのコア資産を用いて派生製品を開発する必要に迫られた。コア組織も常に技術競争への対応に追われており、ナレッジを継承する工数を大きく確保することが難しい。派生組織において、更新されていくコア資産に対するナレッジを効率よく獲得して製品開発する方法の確立が課題となった。

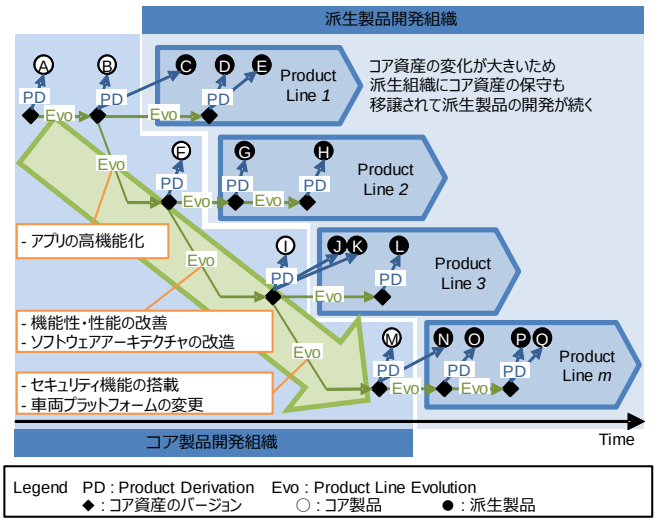


図 1.コア資産の更新と開発組織の対応関係

3.改善策を導き出した経緯

ソフトウェアの開発におけるナレッジの獲得には目的が伴う。現場においては製品を実現することが目的である。製品を実現するためには、実現すべき要求を獲得する必要がある。ナレッジの獲得と要求の獲得には類似性が見出せるのではないかと考えた。素早い要求獲得をキーワードに、関連研究を紐解いて改善策を導いた。

・プロトタイピング…モックなどのプロトタイプを製造して手に取れるモノを共有しながら顧客と対話的に要求を獲得する

・BML ループ…構築・測定・学習のループを回して世の中の反応を得ながら素早く要求を獲得する

・Scrum…Agile 開発のフレームワークで、透明性・検査・適応により改善を重ねてインクリメンタルに要求を獲得する

これらの研究から得られるポイントは、要求者への実物の提示と反復である。要求者の潜在的な要求も引き出し、修正を加えながら要求を獲得する行為を繰り返す。ナレッジの獲得も同様である。何も対象がない状態で何らかの知識や経験結果の出力を要求しても得られることは少ない。ナレッジ出力の梃子となる対象を素早く構築してフィードバックを得ることで、経験も重ねながら効率的に知識を獲得できると仮定して改善策を立案した。

4.改善策の内容

派生組織の開発では、Scrum のフレームワークを導入していたが、要件開発プロセスはウォーターフォールと同様に上流工程・実装工程・下流工程に区切っていた。部分的なプロセス実行では下流工程で得るべきナレッジの獲得が遅くなるリスクが生じる。また、ナレッジの獲得対象は製品の要求仕様や機能仕様だけでなく、設計仕様や検査環境に対しても必要である。これらを鑑みて、以下の改善策を定めた。

- ナレッジ獲得のフレームワークとして Scrum のフレームワークをベースとする

- 素早い検査と適応を可能とするため、スプリント期間は 1 週間とする（従来は 2 週間で開発していた）
- スプリントの中で、製品開発のプロセスは上流～下流で区切らず、定義・構築・検査の 1 サイクルを回しきる
- 設計や製品仕様理解を目的とする場合でも、モデル図や調査資料、検査実施結果などの成果物を作成する
- 毎スプリントの終盤にナレッジを獲得できるように、コア組織の有識者とのレビューを設ける
- 有識者とのレビュー結果を反映して、次スプリントのプランニングをし、繰り返しナレッジを獲得し続ける

5.改善策の実現方法

改善策を、3 つの異なるシーンにおいて実現した。シーンごとに獲得したいナレッジは異なる。各シーンにおける獲得対象と工夫点を示す。

1) 新たなコア資産から最初の派生製品を開発するシーン

このシーンでは、コア資産のアーキテクチャと製品仕様、検査環境の使用方法へのナレッジを獲得したい。そのため、開発要素をできるだけ小さく分割し、スプリントの中で検査の実施まで完了させることを優先した。その場合、回帰テストなど冗長な工数が掛かるリスクがある。そのため、開発要素として独立性の高い要素を分析して選択した。例えば、通信ノードによって影響する機能が限定されている場合がある。該当のノードに対する開発は独立性が高く、先に実装して検査してしまっても、その他のノードに対する実装後の回帰テストは最小限に抑えられる。

2) 派生製品においてコア製品では保証されていない機能を開発するシーン

このシーンでは、コア資産に組み込まれているものの、製品としてリリースされたことのない機能が対象となる。獲得したいナレッジは、該当機能の振る舞いへの理解である。そのため、機能仕様から読み取れる、派生組織の開発メンバが理解した内容での検査仕様を策定し、実ソフトウェアを検査した結果と疑問点を作成して、コア組織の有識者とレビューする方針とした。

3) 派生組織にてコア資産をリファクタリングするシーン（応用例）

このシーンでは、コア資産のリファクタリングをコア組織ではなく派生組織が請け負っている。リファクタリングの目的はより派生製品を開発し易くすることである。獲得対象ナレッジは、リファクタリング対象コンポーネントのアーキテクチャであるが、コア組織のコンポーネント担当が保有するリファクタリングへの要求も獲得したい。このシーンでは各スプリントで、リファクタリング後の設計モデルとその効果を提示し、リファクタリングに掛けるスプリント数に制約を掛ける方針とした。無制約にリファクタリングを志すと、投資対効果を考慮できず、完璧を追及してしまうリスクが生じる。

6.改善による変化や効果

本改善方法による効果として、各シーンで観察された効果を示す。また、シーン 1 において、ナレッジを獲得するために各開発要件を小さく分割したことで、開発工数が大きく掛かったり、開発期間が延長したりしていないかを評価した。

<シーン別の効果>

1) 新たなコア資産から最初の派生製品を開発するシーン

11 スプリント掛けて派生製品を開発した。毎スプリント開発に必要なナレッジが獲得できたが、獲得件数は 13 件であった。その内、要求仕様に関するナレッジは 3 件、設計に関するナレッジは 6 件、検査環境に関するナレッジは 4 件であった。

要求仕様に関するナレッジには、実際に検査を実施することで気づけたナレッジが 1 件存在している。これは、顧客へ打ち上げて調整が必要な案件であった。開発の終盤に遭遇すると、納期遅延に直結していた。

検査環境に関するナレッジは、コア組織では開発初期に問題として顕在化しており、対応方法も常識化されているが、マニュアル化されていないナレッジであった。これら検査環境のナレッジは、検査実施時に遭遇すると、検査の進捗を遅延させ、納期リスクにつながる。

これらのナレッジは、ナレッジがないと停滞を招くものばかりであったが、開発要件を小さく分割していることで、ある要件が停滞しても、ナレッジ不足を解消している間、他の要件を進めることができ、停滞の影響を最小化することができた。このようにして、ナレッジ不足を効率的に補いながら開発を進めることができた。

2) 派生製品においてコア製品では保証されていない機能を開発するシーン

このシーンでは、4 スプリントを掛けて派生製品向けのコンフィグレーションと機能仕様の策定を進めた。

第 1 スプリント…該当機能の検査仕様を策定しながら機能の振る舞いの理解を進めた。機能仕様の記述では読み取り難い振る舞いを有識者に質問して解消することができた。

第 2 スプリント…検査仕様に従い検査を実施して機能の振る舞いの理解を深めた。ソフトウェアの欠陥を 8 件抽出し、内 2 件は実欠陥、6 件は振る舞いの理解誤りであることが判明した。有識者と議論し、欠陥の修正方法と品質保証がし易いコンフィグレーション方法を定めた。

第 3 スプリント…欠陥修正の効果確認と、コンフィグレーションを変更した場合のソフトウェアの振る舞いを検証した。想定通りの振る舞いにはなるが、連動して影響を受けるその他の振る舞いを抽出し、有識者と共有できた。

第 4 スプリント…必要最小限の検査を完了させ、策定した機能仕様を顧客に提案し、承認を得た。

これらの活動を進める中で、コア組織の有識者でも質問されなければ思い当たらなかったソフトウェアへのナレッジを引き出すことができた。機能仕様ベースで仕様を検討していたのでは、開発終盤に仕様の不整合などが発見されることとなり、その場合は緊急対応化し、コア組織の開発コストも大きく取られかねない。コストと開発期間の両リスクを軽減することに成功した。

3) 派生組織にてコア資産をリファクタリングするシーン（応用例）

このシーンでは、5 スプリントを掛けて、コア資産にあるコンポーネントをリファクタリングした。本シーンでは、コア組織と派生組織の相互のナレッジが交流された。コア組織はコア製品を開発中であり、開発中のコア製品への影響を最小限にしたいという要求と、リファクタリング前のコンポーネントの設計理由を派生組織に与えられた。派生組織は、派生製品の開発で得た経験から、頻繁に変更される可変点と、どのような設計にすると保守し易いのかというナレッジをコア組織に与えることができた。毎スプリント両組織で議論することで、定められた期間内に実現したい設計とリファクタリング後のソフトウェアを構築することができた。

スプリントごとに設計への思い違いや、リファクタリング方針の変更（真の方針）が明らかになった。定期的なレビューを設けていないと期間内の完了が見込めなかった。

<改善方法による冗長コスト発生への評価>

改善方法による冗長コストが発生しているかを評価するため、類似の開発規模を有する 2 つのプロジェクトについて、開発工数を分析した。図 2 に適用前のプロジェクトの開発工数の推移を示す。適用前は分析・定義、実装、テストの各プロセスを順番に実行している。1～3 スプリントは、適用後と比較するために提示しており、開発はしていない。つまり、4～11 の 8 スプリントで開発を終えている。図 3 に適用後のプロジェクトの開発工数の推移を示す。適用後は毎スプリント分析・定義、実装、テストのサイクルを回しているため、スプリントによって特定のプロセスに工数が偏ることはない。1～3 スプリントは別プロジェクトの開発を並行しており、小さな開発工数が計上されている。

別プロジェクトとの並行開発期間を均(なら)すと、両プロジェクトはほぼ同等の工数と同等の期間で開発を完了できていることがわかった。改善策適用後でも、冗長なコストや開発期間の遅延が発生していないことを確認できた。

7.改善活動の妥当性確認

Agile 開発の開発フレームワークである Scrum を応用することで、開発に必要なナレッジを効率的に獲得することができた。Agile 開発では、元々学習と反復により素早く改善を繰り返す仕組みを備えている。新規開発における要求や実現方法の不確実性を軽減するだけでなく、何らかの不確実性を備えた要素を含んだ開発であれば、Agile 開発のフレームワークを応用することは有効であると考えられる。

但し、要求獲得もナレッジ獲得も、獲得される内容が不確実であったとしても、獲得するという目的は重要である。何を獲得しようとするのか、という目的を定義せず、何かを獲得しようと、ひとまず何かを作ってみる、という場合は失敗する可能性が高いと思われる。その場合、獲得するナレッジは増えたとしても、実利に結びつかず、活動コストばかり膨れ上がるリスクが生じるため、開発フレームワークだけの適用には注意が必要である。

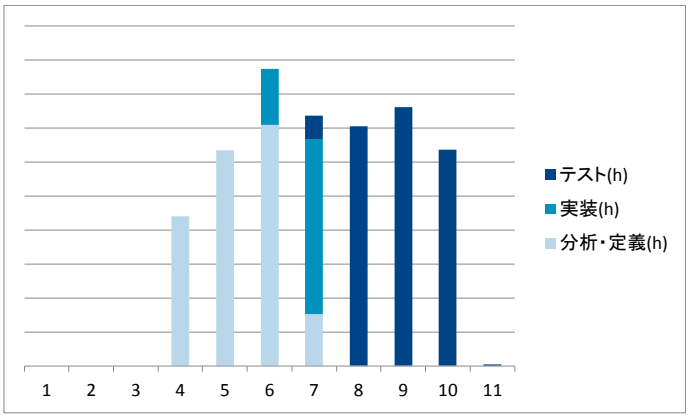


図 2.改善策適用前の開発工数の推移

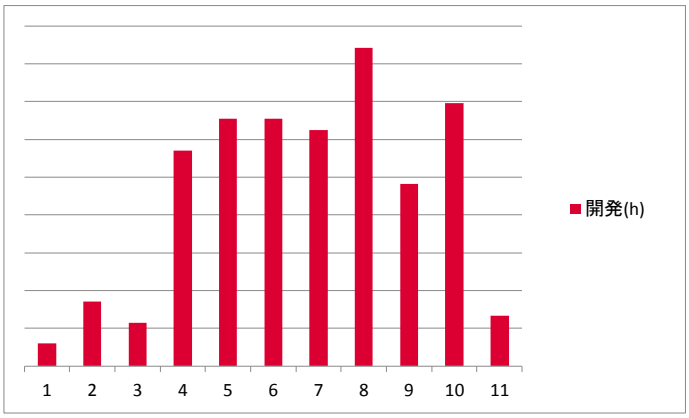


図 3.改善策適用後の開発工数の推移

A. 参考情報

[1] B. W. Boehm, A spiral model of software development and enhancement, Computer, Vol. 21, No. 5, 1988, pp. 61-72.

[2] M. Cohn, Agile Estimating and Planning, Prentice Hall, 2005.

[3] 野中郁次郎, 知識創造企業, 東洋経済新報社, 1996.

[4] エリック・リース, リーン・スタートアップ, 日経 BP 社, 2012.

1B3「アジャイル開発におけるプロセス改善事例 ～楽しく・早く・確実に～」中野安美（ニッセイ情報テクノロジー）

<タイトル>

アジャイル開発におけるプロセス改善事例 ～楽しく・早く・確実に～

<サブタイトル>

<発表者>

氏名（ふりがな）： 中野 安美 (なかの やすみ)
所属： ニッセイ情報テクノロジー株式会社

<共同執筆者>

氏名（ふりがな）：
所属：

<主張したい点>

２０１５年よりアジャイル開発をはじめて以降、チーム運営など様々な工夫を重ねて改善したことにより、チームの自律化、活性化が行えた。我々が工夫した施策と効果について具体的にご紹介させていただきたい。

<キーワード>

アジャイル、スクラム、プロセス改善、効率化

<想定する聴衆>

ソフトウェアエンジニア、アジャイル開発に興味ある方、アジャイル開発経験者

<活動時期>

2015 年 8 月～2017 年 5 月(継続中)

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動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、2015 年 8 月よりアジャイル開発をスタートさせた。当初はチーム全員が初めてだったこともあり、チームの活性化とパフォーマンス向上を行なう必要があった。また、金融系 IT 企業としては品質にも拘る必要があり、品質を確保する必要があった。

2.改善したいこと

主な改善したいことは、次の３点である。

① コミュニケーションの活性化

会話を多くすることにより、チームの意見をまとめるスピードを向上

②品質確保しつつ、テストを効率化

日々更新されるプログラムの品質確保とテスト効率の向上

② プロダクト開発の最適化

１スプリントあたりのパフォーマンス向上

3.改善策を導き出した経緯

改善策を導き出した経緯は、次の４点である。

・各スプリントふりかえり（ＫＰＴ）において、継続的な改善活動を実施

・セミナーや書籍などによる情報収集

・コミュニティへの参加による他社取組み事例の情報収集

・アジャイルコーチのアドバイス

上記のうち、最も大きな効果をもたらしたのは、「ふりかえり（ＫＰＴ）」を実施したことである。ふりかえりでは、次のスプリントで出来る改善策（Ｔｒｙ）をメンバー全員で考え、合意した上で施策を実行した。また、確実に実行するために、スプリント途中でＴｒｙ施策の達成状況も確認しながら進めていった。

4.改善策の内容

主な施策は以下のとおり。

①コミュニケーション活性化

-会話を増やす工夫

-楽しくする工夫

②品質確保、テスト効率化

-テスト自動化、CI ツールの導入、レビュー精度の向上

③プロダクト開発の最適化

-開発者サイドのタスク（技術課題、環境構築）とフィーチャーのトータルで優先順位を判断

-各スプリントで消化するスプリントバックログの目標を２段階で設定（Must ラインと攻めライン）

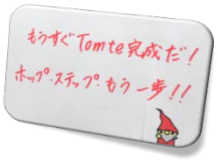
5.改善策の実現方法

① コミュニケーション活性化

(1)会話を増やす工夫

施策項目	施策内容	説明
デイリーミーティング	司会は日直 順番はくじ引き その日に行うタスクを決めてから誰が担当するかを決定 	当初、スクラムマスター（ＳＭ）が司会をしていたが、ＳＭへの報告会のような雰囲気で報告だけして終わるような感じであった。施策後は、開発メンバーが個人よりチーム全体として状況を捉えるようになり、他メンバーの状況を気にするなど、会話の内容や発言量が変化していった。
お菓子神社	時間を決めて休憩タイム 	当初、休憩する時間もまちまちで、チームメンバーよりお菓子神社を有効に活用できていないという意見がでたため、全員が時間を決めてお菓子タイムを設けることとした。全員で話すタイミングができたことにより、情報共有が活発になった。また、立派な鳥居を飾ることにより、楽しさもアップした。
Help カード	悩んだときにカードをあげる 	開発しているときに悩むことは多々あるが、答えも導き出せない状態で一人で悩んでいる時間がもったいないという意見があり、その改善のためにＨｅｌｐカードを作成した。これによりメンバー全員に躓いている状況が認知され、早めに共有できるしくみができ、課題解決スピードの向上と、時間に対する意識改善ができた。

(２)楽しくする工夫

施策項目	施策内容	説明
キャラクター	プロダクトのキャラクター作成 	スクラムマスター（ＳＭ）が「遊び心を取り入れて」をテーマにキャラクターを思いついて作成。開発フェーズや季節イベントごとに表情をかえるなど楽しい雰囲気づくりを行った。
スローガン	スプリントごとにスローガンを決める 	スプリントの開発内容にあわせ、プロダクトオーナーが「スローガン」を決めて、メンバーのやる気アップできるよう工夫をした。（みんなのスプリント毎のお楽しみ）

② 品質確保、テスト効率化

施策項目	施策内容	説明
テストコードを書く	テストツール（ＪＵｎｉｔ、Ｓｅｌｅｎｉｕｍ）の導入	日々更新されるソースコードに対して、デグレードの排除とプログラム品質確保の必要性を感じ、テストツールを導入することとした。 カバレッジ 90%を目標にし、カバーできない部分はレビューで担保することにより品質確保した。ソースコード上のテスト網羅性を確認しながらの品質コントロールが可能となり、品質向上を行うことができた。
テスト自動化、ＣＩツール導入	Ｊｅｎｋｉｎｓ導入によるテスト自動化	日々のテストを自動化することにより、テスト時間の効率化を図った。夜間にテスト実行することにより、出勤時にテスト結果の確認ができ、前日の結果との比較からエラー混入した影響の特定などが以前と比較してスピーディに行えるようになった。
テストケース設計方法の改善	Ｅｘｃｅｌでテスト設計ツールを作成	テストケース設計を行う専用の Excel ツールを作成。JUnit、Selenium が Excel ツールのテスト設計を読み込んで実行できるようにした。表形式でテストケースを表すため、テストコードを書くよりも網羅性の確認などテストケースレビューがしやすく、ケースレビューの質向上が図れた。

③ プロダクト開発の最適化

施策項目	施策内容	説明
バックログの優先順位判断	開発者サイドのタスク（技術課題、環境構築）とフィーチャーのトータルで優先順位を判断	スプリントプランニングの際に、フィーチャーの優先度と開発サイドのタスクの優先度について、プロダクトオーナーと開発メンバー間で相談し、プロダクト全体の効率的な進め方を全員で話しながら優先順位づけした。
各スプリントの消化目標とするバックログの設定	各スプリントで消化するバックログの目標を 2 段階で設定（Must ラインと攻めライン）	各スプリントの成果を最大にするため、目標とするバックログを 2 段階で設定した。「Must ライン」は、開発メンバーから判断して確実に出来そうなライン。「攻めライン」は、少し厳しい目標の達成水準ライン。 これにより、高い目標を意識してタスクを進めることにより気持ちの緩みによるロスをなくすことができるとともに、開発メンバーにとっては Must ラインを超えていることで上回り達成感も得られた。

6.改善による変化や効果

① コミュニケーション活性化

・気づきの発信やチーム全体の状況理解など、チーム全体でゴールを目指すという意識が醸成された

・悩みをかかえてじっと考えることがなくなり、時間に対する意識が変わった

・継続的なプロセス改善を行うことができた

・楽しみながら仕事をする風土の醸成(モチベーションアップ)

② 品質確保、テスト効率化

・テストコードを書くことが定着化

・プログラム品質の確保

・テストケースレビューの質向上

・リファクタリングが容易にできるテスト資産を蓄積

③ プロダクト開発の最適化

・プロダクトオーナーと開発者間でなんでも話せる雰囲気、信頼感の醸成

・タスク順序を両者納得感をもって決めることにより、プロダクト開発において全体最適なステップで開発

7.改善活動の妥当性確認

(今回の改善取組の妥当性確認)

・今回の各種取組を通じ、チームとして効率的に成果を出していくこと、またチームとしてアウトプットを出すためにメンバーを補い合う意識が醸成された。これにより、早期に課題共有を行い、仕様調整や課題解決の時間短縮を行なうことができた。

・テスト工程での品質確保とリファクタリングがしやすい環境が整い、プログラム品質向上とテスト工数削減を実現した。

・機能とコストのバランスがとれたプロダクト開発を行なうことができた。

・ベロシティ計測の観点から日々の業務終了時間を決めていたが、結果的に残業抑制になり、「働き方改善」に繋がった。

(今後の改善施策)

・チーム間のプロセス改善施策を共有し、よい施策の横展開。

・メトリクスによる定量分析の実施。メトリクスデータを継続的に収集し、複数の要素を統合的に評価して異常の検知やチームの改善施策などに繋げる。

・当社に即したアジャイル開発の開発規約、運営ルールを整備、定量的な判断基準などを策定する。

・ウォーターフォール開発チームでも有効と思われるプラクティスを導入し、プロセス改善意識の醸成を行なう。

1B4「アジャイル推進活動をアジャイルにやってみた」伊藤裕子（東芝）

<タイトル>

アジャイル推進活動をアジャイルにやってみた

<サブタイトル>

SPI 活動もアジャイルに

<発表者>

氏名（ふりがな）：伊藤裕子(いとうゆうこ)
所属： 株式会社東芝　ソフトウェア技術センター プロセス・品質技術開発部

<主張したい点>

アジャイル普及展開活動にスクラムを適用した結果、継続的な改善が行われ、スクラムのメリットを享受できた

<キーワード>

アジャイル、スクラム、SEPG、SPI、ソフトウェア以外の開発でスクラム、教育コンテンツの開発、アジャイル推進活動、展開施策、ふりかえり、KPT、スプリント、スクラムマスタ

<想定する聴衆>

SPI 推進者、アジャイル推進者

＜活動時期＞

継続中

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
☑ 改善活動を実施したが、結果はまだ明確ではない段階
☑ 改善活動の結果が明確になっている段階
☑ その他（活動は継続しているので、結果が明確になっている部分となっていない部分がある）

<発表内容>

1.背景

東芝グループ内には、コーポレートの立場で SPI を推進する組織（コーポレート SEPG）があり、製品開発部門への支援や教育の実施などを行っている。2015 年度から本組織内に本格的に東芝グループ内の製品開発へのアジャイル普及展開のミッションが加わり、アジャイル推進チームとして活動をしている。アジャイル推進チームの活動は約 2 年半が経過、発表者は推進チームに加わって 2 年目。チームのメンバの入れ替えなどもありながら、現在は 6 名でアジャイルの普及展開を進めている。

2.改善したいこと

アジャイル推進チームのメンバは 2015 年当時 4 名。スクラム開発経験者は 1 名のみ、チーム 4 名中 3 名認定スクラムマスタだが実プロジェクトへの適用については経験不足だった。これから増えていきつつある支援部門に対し、実経験をもとにしたアドバイスができないことが弱みだった。社内の宣伝活動の結果、適用部門数が増えてきたが、推進チームメンバの増員は難しい状況で支援の質を保つことが難しくなっていた。

アジャイル推進チームでは、アジャイル普及展開のために、製品開発部門の支援、教育の実施、その他普及展開に向け必要な活動をすることが求められていた。アジャイル開発支援のニーズにこたえるために、教育コンテンツの早期リリースや、コミュニティの早期立ち上げは必須であり、東芝グループ内の有識者や支援部門からの意見を反映させながらタイムリーに全社施策を展開していく必要があった。

3.改善策を導き出した経緯

アジャイル推進チームでは、支援部門数の拡大に伴い教育コンテンツの開発を最優先とする方針をたてた。また、教育コンテンツの開発自体をスクラムで開発することで、アジャイル／スクラム開発の経験を積むこととした。教育コンテンツの開発は、2015 年 7 月から始めた。チーム初めてのスクラムであったため、基本に忠実に、スクラムを実施。1 スプリントは 1 週間、3 か月で 9 スプリントを実施し、出荷可能な成果物(教育コンテンツ開発では、それだけで説明できるパワーポイントスライドと定義した)を作り上げた。その結果、教育コンテンツである、ソフトウェア以外の開発でスクラムのプラクティスを活用した開発を進めることができ、メンバ全員がスクラムを体験できた。また、結果的に従来のスタイルに比べ短い期間で教育コンテンツを開発することができた。ソフトウェアに限らず、スクラムを適用できることが実体験できた。

教育コンテンツ開発はアジャイル／スクラムを適用し進めていたが、アジャイル推進活動全体としては、従来の通り中長期計画を立てて、その計画に沿って進めていた。しかし、推進活動全体にもアジャイルを適用できる可能性がある。教育コンテンツの開発に味をしめたアジャイル推進チームは、2015 年 10 月から、アジャイル普及展開のための推進活動全体もスクラムで進めることを検討した。推進活動は PJ と異なり有期ではなく継続的な活動。推進活動にアジャイルを適用することで適用範囲が広がることに気付いた。また、アジャイル推進チームでは、メンバの入れ替わりがあり、スクラムを体験していない新規メンバもいた。スキルアップは常に必要な状況だった。

4.改善策の内容

2015 年 10 月からアジャイル普及展開のための推進活動全体を、スクラムのフレームワークを活用し、アジャイルに進めた。アジャイル推進のための全社展開施策をプロダクトバックログとし、推進活動のリーダがプロダクトオーナの役割を担い、全社展開施策の優先順位づけを行った。アジャイル推進チームが開発チームとなり、その中にスクラムマスタの役割を置いた。

5.改善策の実現方法

全社展開施策をチームメンバ全員で挙げ、初期プロダクトバックログを作成した。プロダクトバックログには、教育開発に関わるものの他、「各製品開発部門でアジャイルに興味ある人を集めてコミュニティを立ち上げたい」、「アジャイルに関するイベントを開催したい」、「支援の際に活用できるツールセットを作りたい」など全社展開施策についてのさまざまなアイデアが挙がった。プロダクトバックログは見える所に貼っておき、常に共有されている状態にした。節目を 3 ヶ月ごとに設けてプロダクトバックログの見直しを行い、

追加のアイデアを取り入れ活動の優先順位を見直した。全社展開施策のプロダクトバックログの中には教育開発や部門支援、コミュニティの企画など複数の活動が含まれている。スプリントは 2 週間とし、スプリントバックログ(この 2 週間の期間にやるべき施策と、その施策を実現するために分解されたタスク)を作った。デイリースクラムも実施。毎朝 09:30～09:45 に継続して実施している。スプリント毎に KPT によるふりかえりを実施している。

6.改善による変化や効果

アジャイル推進活動をスクラムのフレームワークに沿って進めたことで、アジャイルの普及展開のための施策を、その時々の要望の変化に対応しながら、優先順位をつけて実施することができた。全社向け教育の立上げ、イベントの立上げ、実践者が集まる場の立上げ、支援キットの構築、これらを製品開発部門の支援件数が拡大していく中実現することができた。プロダクトバックログ項目をアジャイル推進チームメンバ全員で洗い出したため、全社展開施策に対するオーナーシップが上がり、自律的になったためと考えられる。

また、アジャイル推進チームメンバ自身のアジャイル支援スキルが向上した。スクラムの会議体(スプリント計画、ふりかえり、スプリントレビュー、デイリースクラムなど)を実施することで、進め方を体験することができた。この結果、製品開発部門支援において、自分たちが実際に適用した経験も踏まえながら、アドバイスできるようになった。

スクラム適用が特に有効であったと考えられる点を 3 点挙げる：

1 つは、スプリントに区切られているため活動の推進力が生まれた。全社展開施策の中には、コミュニティの運営など長期にわたるものがある。プロダクトバックログには優先順位がつけられており、優先順位に応じて直近のスプリントに割り当てていく。これにより、必要なタイミングで取りこぼしなく進めることができるようになった。

1 つは、ふりかえりを短期間でくりかえし実施することで、チーム内で気付いた問題点に対し、早期に改善のフィードバックをかけられるようになった。スクラムでは、ふりかえりの時間は定期的にあるため、改善について考える時間が必ずある。チーム内の細かいやり方はふりかえりを通して、チームがやりやすいように改善されていった。

1 つはスクラムマスタの役割をチームメンバの中に置いたことで、改善の芽を早めに発見できるようになった。スクラムマスタは、チームが効率的に動けるように支援する役割を持つ。チームメンバはどうしても目の前の作業に注力しがちになるが、スクラムマスタは少し先まで見通した上で、効率が悪い面がないかを常に探し続ける。今回の例では、スクラムマスタが、スプリントバックログのカンバンを使う環境を早期に Redmine で構築したことで、チームを早く軌道に乗せることができた。また、チームがスプリント期間内に予定していたタスクを完了できないことに気付いたスクラムマスタが、バーンダウンチャートを書き毎日更新したことで進捗が見える化され、チームメンバも進捗を気にしてチーム内で他の人のタスクを手伝う／取るなどの変化があった。

7.改善活動の妥当性確認

改善活動の狙い通り、アジャイル推進活動をスクラムのフレームワークに沿って進めることができた。状況変化への柔軟な対応や、チーム内の自律性・モチベーションの向上などのアジャイル／スクラムの効果を得ることができた。以上のことから、ソフトウェア以外を対象とした開発、かつ、継続的な活動においても、アジャイル／スクラム適用は有効であると考えられる。

ただし今回は成功要因として、アジャイル推進チームのメンバ全員が、SPI 推進については 3～10 年以上の経験をつんだ熟練者であったことが挙げられる。アジャイル適用は本質的には SPI 活動であり、部門の課題に適応した解決策を提案・実践していくためには、アジャイル／スクラムなどの固有知識だけでなく、SPI の知識が必須である。これは、アジャイルをソフトウェア開発に適用した場合に、メンバにソフトウェア開発の基本スキルが求められるのと同様である。つまり今回の改善活動では、アジャイル／スクラムの効果が発揮しやすいベースが存在したことは付記しておく。

A. 参考情報

[1] アジャイル開発の教育コンテンツ作成をアジャイル開発で挑戦～ポンコツ PO と指示待ちチームメンバがスクラムに取り組んだ 3 ヶ月間の記録～, 伊藤裕子, アジャイルジャパン 2016 プレイベント企画「初心者向けセミナー」(2015), http://www.agilejapan.org/event.html

1C1「異なる開発組織における継続的統合に向けた Queuing 統合の提案と適用」白木徹（デンソー）

<タイトル>

異なる開発組織における継続的統合に向けた Queuing 統合の提案と適用

<サブタイトル>

－

<発表者>

氏名（ふりがな）：白木 徹（しらき とおる）
所属： ㈱デンソー

<共同執筆者>

氏名（ふりがな）： 林健吾（はやし けんご）
所属： ㈱デンソー

<主張したい点>

GUI アプリケーションなどのソフトウェアシステムは，MVC (Model View Controller)アーキテクチャパターンを採用している．MVC の各サブシステムを開発する組織が異なっていることからビッグバン統合を採用しているが，結合テスト期間の不足を招いている．MVC の中で View の開発量が多い点と，View のコンポーネントの独立性が高い点に着目し，異なる組織間での継続的統合に向けた Queuing 統合を提案・導入して品質リスクを低減する．提案プロセスを自動車の HUD ソフトウェア開発へ適用し，開発効率と品質の向上を実現した．

<キーワード>

GUI、MVC、インクリメンタル開発、継続的統合、CI、自動車システム

<想定する聴衆>

ソフトウェアエンジニア、組込系技術者、SEPG

＜活動時期＞

2016 年 12 月～2017 年 4 月（以降、継続中）

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（ ）

<発表内容>

1.背景

発表者らは，運転者に情報を提供する手段としてフロントガラスに情報を投影するヘッドアップディスプレイ(HUD：Head Up Display）のソフトウェア開発に従事している．自動車システムの複雑化と搭載車両の拡大によって表示コンテンツ数が急増し，ソフトウェア開発量も急増している．HUD ソフトウェアは GUI 開発が中心となり，ユーザへの表示コンテンツが多様であるので，MVC（Model View Controller)アーキテクチャパターンを採用している[2][5]．MVC の 3 つのサブシステムの開発は必要となるスキルやツールセットが異なり，開発組織もサブシステム毎に異なる部門が開発を分担している．分担開発を行っていることから，システム試験のためのソフトウェア統合方法としてビッグバン統合（以下，BI）を採用していた．主体はソフトウェアの開発組織であり，発表者は主体のリーダを担う．増大するソフトウェア開発量に対して危機感を感じ，活動を開始した．

2.改善したいこと

開発量の拡大に伴い，システム試験における工数は増大する．特に，View の仕様変更の頻度が高く，ソフトウェア変更後の妥当性確認はソフトウェアの結合後にしか行えない．BI では開発量の増大と共に統合時期が開発プロセスの終盤に遅延する傾向があり，品質保証の工数と期間が確保できなくなる．ソフトウェアの統合を早期化し，システム試験を早期に着手して品質保証期間を確保することで，品質低下リスクと納期遅延リスクの低減を目指す．

3.改善策を導き出した経緯

BI の問題を解消するためには，継続的統合（CI）に移行すればよい[1]．しかし，MVC の各サブシステムは異なる組織で異なる開発速度，開発粒度で開発しているため，統合すべきコンポーネントの完成タイミングが異なる．これを解決する方法は，単純には以下 2 つの方法が考えられる．

A. 各サブシステムのコンテンツ開発の同期をとり，統合タイミングに合わせる
⇛開発速度が速いサブシステムで待ちが発生して開発効率は悪化する

B. 各サブシステムのコンテンツ開発が完了したら，順次統合していく
⇛不整合のある組み合わせが生じるなど，不具合が無秩序に生じて品質確保が困難になる

これらの方法の欠点を分析することで，統合時の不具合のスコープは限定されるが，各サブシステムの開発速度を妨げない統合方法を考案すればよいと考えた．

4.改善策の内容

統合を同期せず，サブシステムごとに開発を進めることのできる統合方法として Queuing 統合（以下，QI）を提案する．QI の例を図 1 に示す．QI では，統合して機能を実現するサブシステムは，開発順序だけ同じくし，開発完了のタイミングは同期させなくてもよい．統合の同期を取らないことで開発の待ちが生じず，各サブシステムの開発効率を低下させない．開発完了したサブシステムは，対象コンテンツごとに統合対象のキューに加えられる．特定のコンテンツを実現するすべてのサブシステムがそれぞれのキューの先頭に揃ったところで，システムとして統合し結合テストを開始する．統合するサブシステムのスコープを限定することで，結合テストで発見される不具合を解析するスコープも限定でき，統合後の工数も最小化することが可能となる．

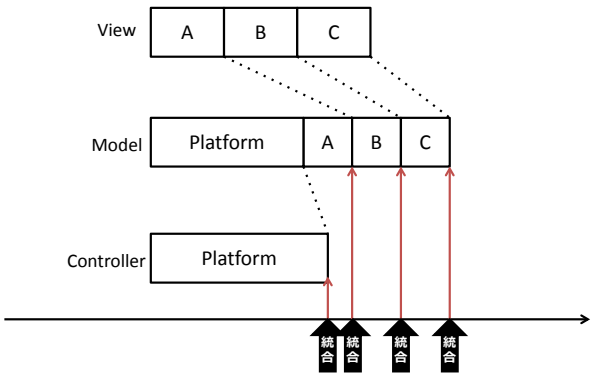


図 1 Queuing 統合の例

5.改善策の実現方法

提案した開発方法を，HUD の実開発に適用する．QI を導入する場合，理想的には各サブシステムの開発には同期を持たせず，柔軟なタイミングで統合できることが望ましい．しかし，非同期な開発と柔軟な統合は，プロセスの変化が大きく現場の混乱を引き起こす．よって，QI のメリットは最大限に発揮できないが，容易な導入を促すため，アジャイル開発と同様に固定期間を開発リズムとするスプリント[3][4]を導入し，スプリントごとに開発対象を計画して実現した．スプリントを導入することで，統合日を周期的に設定でき，統合のための調整コストを下げ，統合用のツール導入コストを下げられる．スプリント期間を 1 週間に設定し，12 スプリントの計画を立案した．

導入の混乱を低減させるため，開発の優先順位として下記ルールを設定した．優先度が高い順に示す．

(1) 開発 Platform，Controller を優先して開発する（すべての開発の土台をはじめに）．
(2) 開発が容易な機能から開発する（新しい導入に慣れるためには易しいものから）．
(3) 移植など変更規模が少ないコンテンツから開発する．
(4) 変更開発，新規開発となるコンテンツを開発する．

これらの開発の優先順位に従って作成した開発計画を図 2 に示す．図のように各サブシステムの Platform の開発を先行して行い，それ以降は各コンテンツをインクリメンタルに開発していく．このようにすることで開発が終わったコンテンツは次のスプリントで評価を実施することができる．例えば，App1 に対して，View ではスプリント 1，Model ではスプリント 3 で開発が完了しているため，スプリント 3 の完了時点で統合ができ，スプリント 4 で評価をすることが可能となる．

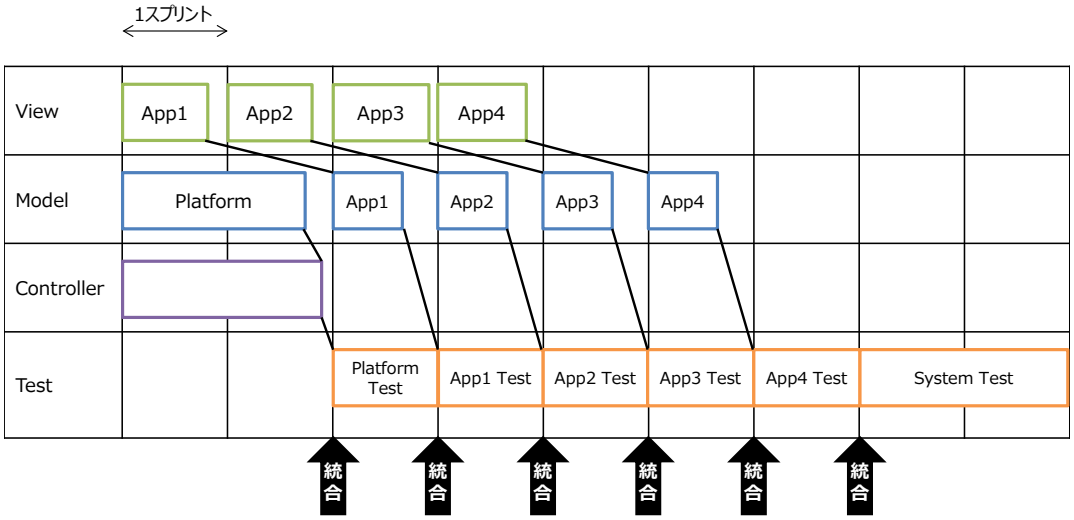


図 2 開発計画の例

6.改善による変化や効果

比較対象は同一の開発車両，同一の開発メンバーにおける，提案方法の適用前とする．但し，開発規模は異なっている．単純に CI に移行しようとした場合にデメリットとして生じる、品質確保と開発効率について評価した．

＜品質確保向上効果＞

図 3 に提案方法の適用前におけるバグ曲線と不具合の内訳を示す．適用前では，不具合の検出時期が納入の 2 週間前であることが分かる．また，不具合の内容は，単体テストが多くを占めており，ソフトウェアの改修期間が十分に確保できていないことが分かる．図 4 に提案方法の適用後におけるバグ曲線と不具合の内訳を示す．適用前に比べて，不具合検出が早期化できていることが分かる．スプリント 2 でリグレッションテスト 5 件，スプリント 5 以降では毎スプリントで単体テストの不具合を検出し，計 11 件検出できていることが分かる．不具合総数 23 件中 16 件はソフト不具合であり，ソフトウェアの改修が必要となるが，不具合検出が早期化されたことで改修期間を確保できている．また，不具合の検出方法は，適用前よりも多く，各アクティビティの不具合検出能力が十分に発揮できており，品質が向上できている．以上のことから提案方法を用いることでソフトウェアを早期改修することができ，納期の達成と品質の確保ができた．

＜開発効率向上効果＞

図 5 に適用前の月毎の開発工数の推移と内訳を示す．適用前では，8 月と 9 月に要求分析から設計・製造の大部分を行い，9 月と 10 月で検査を実施していることが分かる．また，全体の開発工数では，検査を多く実施している 9 月が最も多く，開発の後半に検査工数が集中している．図 6 に適用後の月毎の開発工数の推移と内訳を示す．適用後では，月毎の要求分析・設計・製造・検査の工数が平準化され，全体の開発工数においても平準化されていることが分かる．

以上のことから，全体の開発において，設計者・実装者・テスト/評価の担当者の負荷が平準化され，開発効率が向上した．

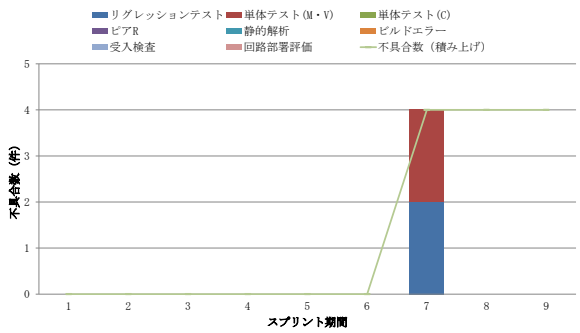


図 3 バグ曲線と不具合の内訳（適用前）

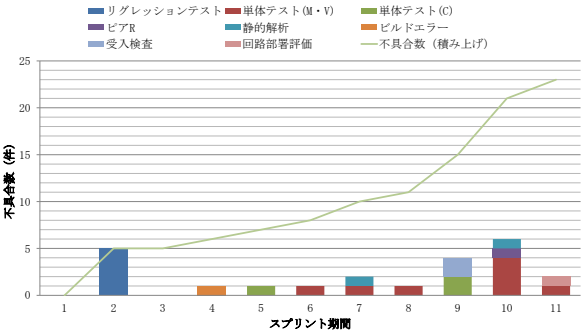


図 4 バグ曲線と不具合の内訳（適用後）

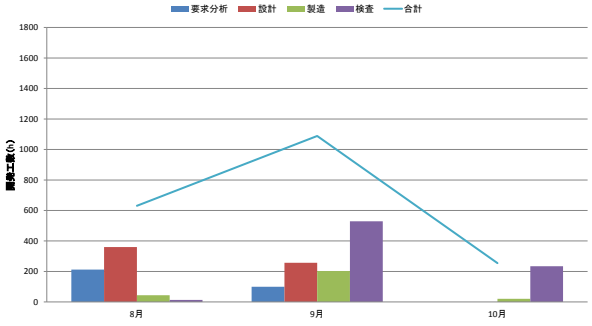


図 5 月毎の開発工数の推移と内訳（適用前）

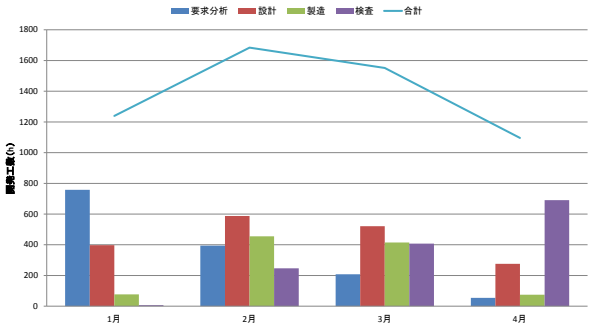


図 6 月毎の開発工数の推移と内訳（適用後）

7.改善活動の妥当性確認

図 4 の内訳の分析を進めるとスプリント 9，10 で多くの不具合を検出していることが分かる．原因を調査したところ，スプリント 9 では，View 層の単体テストにおいて，それ以前のスプリントとは異なる観点での単体テストをすべてのコンテンツに対して実施した結果であることが分かった．スプリント 10 では，顧客からの急な仕様追加に対応した結果発生したものであった．

以上のことから，開発プロセスの詳細をより厳密に規定し，別観点の単体テストを View の各コンテンツの単体テストと同じスプリント中に実施しておけば，ソフトウェア改修の更なる早期化を実現することができ，さらに効果を高める余地があることがわかった．また，スプリントによって開発を計画した副次的効果として，開発期間が一定期間で区切られることで，誰が何をいつまでに開発するのかが明確になり，開発チームの開発能力の把握と，開発チーム全体としてのリソース管理がしやすくなった．その結果，スプリント 10 で発生した急な仕様追加にも柔軟に対応することができた．BI ではソフトウェアの完成度も開発状況も把握することが困難であり，納期リスクが顕在化する可能性が高い．

一方，BI からインクリメンタル開発に移行した結果として，開発スケジュールの管理や開発メンバーへの周知に対して課題が残った．開発がスケジュール通りに進まない場合には，スケジュールの見直しや人の割当を都度実施する必要があり，開発メンバーとの密なコミュニケーションが欠かせず，管理コストが増加した．また，今回の施行では導入コストを抑えるため，統合タイミングだけでなく、MVC の各サブシステムの開発もスプリントに区切っている．その結果，QI の効果は最大限には得られていない．管理コストの増加を低減し，QI の効果を最大限発揮するために，適切なツールの選定やより詳細なプロセスの策定が必要であ

る.

A. 参考情報

[1] J. Bosch (ed.), Continuous Software Engineering, Springer, 2014.
[2] F. Buschmann, et al., A System of Patterns: Pattern-Oriented Software Architecture, Wiley, 1996 [金澤 典子, ほか (訳), ソフトウェアアーキテクチャ: ソフトウェア開発のためのパターン体系, トッパン, 1999].
[3] 林 健吾, 他, プロダクトライン開発のための反復型プロセスモデルと管理方法の提案と適用評価, SES 2016 論文集, 情報処理学会, Aug. 2016, pp. 195-202.
[4] 前川 直也, 他, わかりやすいアジャイル開発の教科書, SB クリエイティブ, 2013.
[5] R. N. Taylor, et al., Software Architecture, Wiley, 2010.

1C2「高信頼アーキテクチャ設計手法 ATAM の実践」藤原啓一（三菱スペース・ソフトウエア)

<タイトル>

高信頼アーキテクチャ設計手法 ATAM の実践。

<サブタイトル>

いざというときの安全性を担保してくれる信頼性の高いアーキテクチャを作るために。

<発表者>

氏名（ふりがな）：藤原啓一（ふじわらけいいち）
所属： 三菱スペース・ソフトウエア株式会社

<共同執筆者>

氏名（ふりがな）：
所属：

<主張したい点>

派生開発を何度経験しても、新規開発のアーキテクチャ設計ができるようにはならない。度重なる仕様変更・追加が重なると、アーキテクチャが崩れ、新規開発もしくはリファクタリングをしなければならない時期が早晩来るが、派生開発しか経験していない者が行うアーキテクチャ設計は、脆弱なアーキテクチャになる可能性が大きい。「システム換装や製品の大幅機能追加として、新規に近い開発をする度に、必ずプロジェクト崩れを起こす」という原因の一つはここにある。それを予防する一つの手段として、アーキテクチャ設計手順を理解する人を増やし、アーキテクチャ設計に必要な経験情報・知識を組織知として蓄積しておくことが考えられる。その実践的手段として、設計手順は簡易 ATAM（Architecture Trade-off Analysis Method）、経験情報・知識としてはアーキテクチャ・スタイルとする例を紹介する。この手順は、機能安全の設計手法としても問題なく使える。

<キーワード>

アーキテクチャ設計、機能安全、ATAM、派生開発、ユースケース、USDM、EARS

<想定する聴衆>

・システム開発、ソフトウェア開発おけるアーキテクチャ設計に苦労している方。

<活動時期>

2009 年 1 月～ 継続中

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（ ）

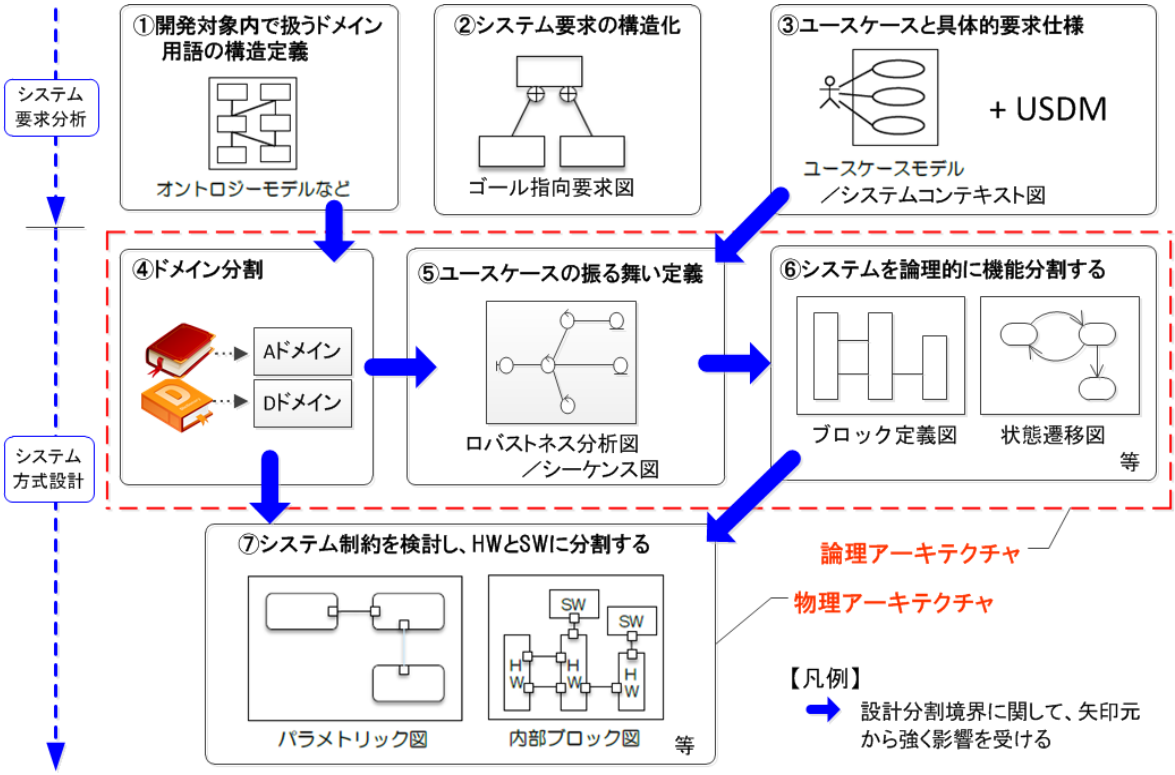
<発表内容>

1.背景

弊社は、宇宙・防衛から民需製品（例：車載機器制御、通信機器制御等）まで、多分野にわたるソフトウェア開発を行っている。どの分野も、最近、派生、流用開発が多くなっており、スクラッチ設計能力を実践の場で磨く機会が少なくなっている。一方、最近、依頼の多い、他社が開発したソフトウェアの改善・改修は、設計段階からのリファクタリングを含むものであり、スクラッチ設計能力が必要とされる。特に、アーキテクチャ設計は、最も難しく、生産性・品質に大きく影響するため、この部分を含めた設計手順の整備は、組織の技術力を高めるだけでなく、製品としての安全性を担保するために必須である。本発表では、弊社で実践しつつ改善を続けている、システム要求仕様作成～ソフトウェア方式設計までの標準設計手法の内、アーキテテクチャ設計に関する概要を紹介する。

2.改善したいこと

現時点での大枠の形として、下記のような標準システム設計手法を整備している。



これまで不足勝ちであった“アーキテクチャ設計”を充実させるために、誤解を恐れず、「非機能要求を実現する構造・振る舞いがアーキテクチャである」と断定して、その記述内容の定義・作成手順化を行い、かつ、機能分解の設計書（機能方式設計書：上図の⑤）とは独立した設計文書（基盤方式設計書：上図の⑥⑦）として作成することで、記述内容のレベル統一を図ってきたが、基盤方式設計書の内容レベルの統一が思うように図れなかった。

また、詳細設計者が、アーキテクチャ設計の不備・不足だと思いつつ、その不備を申し出ずに生じたプログラミング不具合の内容を元に、今までよりも、アーキテクチャ設計レビューで不具合を出し易くなるアーキテクチャ設計書の内容構成や設計手順にする改善が必要であった。

3.改善策を導き出した経緯

あるプロジェクトの要求仕様作成に係わっていないアーキテクチャ設計有識者に、規定の設計手順プロセスに従って、アーキテクチャ設計書作成を作成してもらったが、想定基準の記述レベルに達しなかった。弊社基準のアーキテクトであっても、その時点での、アーキテクチャ記述目次テンプレート及び記述定義では、非機能要求の整合性をとりつつ、根拠をもってアーキテクチャ設計（基

盤方式設計）を作成することは難しいということが分かった。原因は、①アーキテクチャ設計書の記述内容定義に曖昧性が残っていること、②アーキテクチャ設計は、設計根拠を中心にレビューするが、その設計根拠資料を記述する内容定義、具体的手順が無かったことである。そして、このような設計に直接関係する改善が停滞していた間接的要因には、派生開発が多くなりアーキテクチャ設計力を必要とする場面が少なくなっていたこと、管理プロセス改善で組織品質がある程度まで改善していたこともあると考え、現在は設計力強化に改善の重点を移している。

4.改善策の内容

全設計プロセスにわたり試みてきた改善例の一部を下記に示す。発表では、アーキテクチャ設計に絞って説明する。

(1) 要求分析

① 用語定義辞書を準備し、全設計書で使用用語を統一することを必須とした。作業効率化の為に、全設計書に渡って定義されていない用語を検出するツールを準備した。

② 業務系の要求仕様として、業務フロー図、及びユースケースと USDM（Universal Specification Describing Manner）のペアを使うことにとしていたが、組込開発では、業務フローの替わりに、製品仕様からユースケース抽出ができるようゴール指向ベースの要求図を使うこととした。

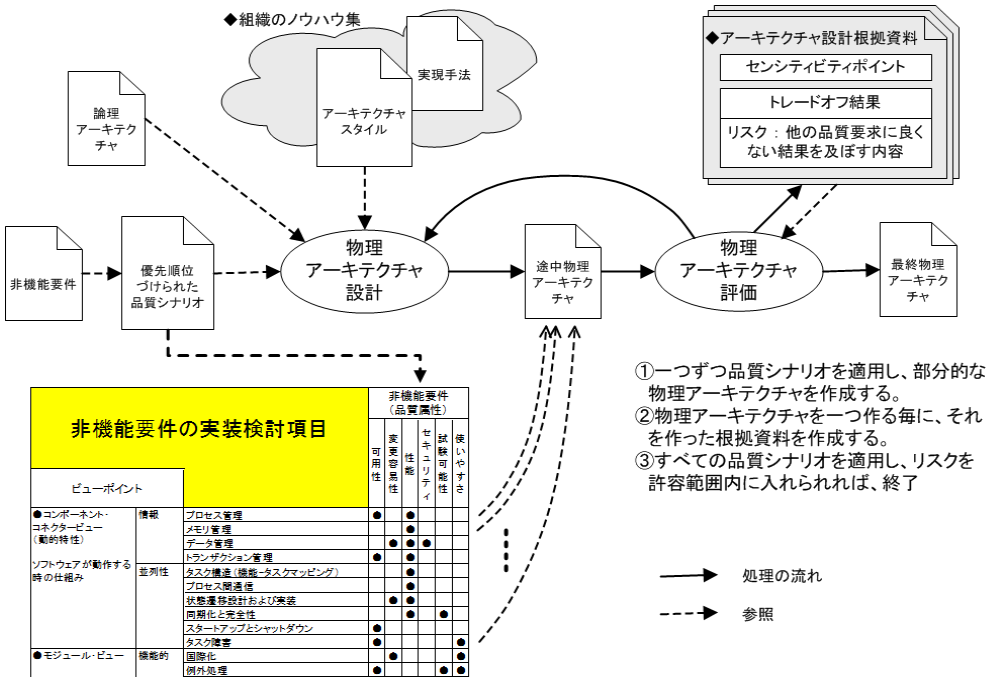
③ USDM 内の仕様の書き方を構造化できるよう EARS(Easy Approach to Requirements Specification)をベースに拡張した形式で記述するようにした。

(2) アーキテクチャ設計

① ドメイン分割基準の一つに、用語の構造分類を用いるようにした。

② どのプロジェクトも行っている機能分割（機能方式設計書）をインプットとして論理アーキテクチャ（初期基盤方式設計書）を作成し、その論理アーキテクチャから物理アーキテクチャを作ることで、最終的な基盤方式設計書を完成させる（標準システム設計手法：④⑥⇒⑦部分）。

③ 論理から物理に変換する際に設計根拠資料を作ることになるが、その手順には、ATAM（Architecture Trade-off Analysis Method）をテーラリングした方法を採用する（下図）。



非機能要件の実装検討項目

ビューポイント			非機能要件（品質属性）					
			可用性	変更容易性	性能	セキュリティ	試験可能性	使いやすさ
●コンポーネント・コネクタービュー（動的特性） ソフトウェアが動作する時の仕組み	情報	プロセス管理	●		●			
		メモリ管理			●			
		データ管理		●	●	●		
		トランザクション管理	●		●			
	並列性	タスク構造（機能−タスクマッピング）			●			
		プロセス間通信			●			
		状態遷移設計および実装		●	●			
		同期化と完全性			●		●	
		スタートアップとシャットダウン	●					
		タスク障害	●					●
●モジュール・ビュー	機能的	国際化		●				●
		例外処理	●				●	●

5.改善策の実現方法

アーキテクチャ設計の改善では、次の４つを準備し、いくつかのプロジェクトに適用し、設計結果が分かりやすくなるかを確認した。

①論理アーキテクチャから物理アーキテクチャへの変換手順書
②アーキテクチャ設計根拠記述フォーマットの定義
③アーキテクチャ・スタイルの準備
④基盤アーキテクチャ設計書の記述内容定義の改訂

6.改善による変化や効果

- アーキテクチャレビューの質向上
  アーキテクチャ設計書の作成過程において品質要求として競合している点とその解決内容が明確になり、レビューが容易となった（それまでは、作成者がレビュー観点を準備してレビューをしてもらう形態であった）。
- アーキテクト育成
  アーキテクトのスキル向上の指標になった。これまでアーキテクトと称されてきた人でも、実際には、どのような根拠で、その設計にしたのかを説明できない部分（ギャップ）がレビューを通じて明確になり、そのレビューシートを例に教育フィードバックできるようになった。また、それまでアーキテクト一歩手前と言われている人でも、この方法であれば、アーキテクチャ設計書が残せることが分かった。
- 機能安全にも耐える設計手法であると考えている。
  本手法に特段の追加修正を行うことなく、そのまま機能安全設計手法に利用できる（ただし、検証手法は別に設ける必要あり）。

7.改善活動の妥当性確認

- 費用対効果
  設計根拠資料を作成する分、設計書作成作業が増えているはずだが、レビューも含めた全体設計作業時間は増えておらず、設計記述の充実が図れ、開発の後段での誤りが少なくなり、また、保守性が高まった分だけ、費用対効果が確認できる。

A. 参考情報

[1] 要求から詳細設計までをシームレスに行うアジャイル開発手法 藤原啓一 MSS 技報（2014）
[2] 大規模組込システムの要求分析、システム方式設計、そして、ソフトウェア設計までをつなぐモデルベース設計手法 藤原啓一 MSS 技報（2015）
[3] 要求工学・設計開発技術部会 非機能要求とアーキテクチャ WG2006 年度報告書 IPA SEC 編
[4] ソースコード主体の派生開発からモデル主体の派生開発へ 派生開発推進協議会 T20 研究会（2013.5.24）
[5] 設計力強化全社活動 - アーキテクト育成とスキル認定 – SPI Japan 2011 パナソニック株式会社（2011）
[6] プロセス分析に基づくドキュメント再構成によるプロセス改善 SPI Japan 2016 株式会社デンソー（2016）

1C3「プログラム設計要否判定による工数削減」丹羽郁美（住友電工情報システム）

<タイトル>

プログラム設計要否判定による工数削減

<サブタイトル>

<発表者>

氏名（ふりがな）：丹羽　郁美（にわ　いくみ）
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　第二システム部　第三システムグループ

<共同執筆者>

氏名（ふりがな）：
所属：

<主張したい点>

プログラム設計の中で付加価値を生まない作業をなくすことで、コスト削減を目指す取り組みについて述べる。

<キーワード>

プログラム設計、コスト削減

<想定する聴衆>

コスト削減に取り組んでいる方、プロセス改善推進者、プロジェクトリーダー、システムエンジニア

＜活動時期＞

2014 年 10 月～　継続中

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
■ 改善活動を実施したが、結果はまだ明確ではない段階
□ 改善活動の結果が明確になっている段階
□ その他（　　）

<発表内容>

1.背景

当組織では中期ビジョンとしてコスト削減目標を設定している。その実現の為にワーキンググループを立ち上げてコスト削減策を検討することになった。ワーキンググループでは外部設計から統合テストまでの開発プロセスの中で付加価値を生まない作業があるのではないかと考えた。特にプログラム設計は外部仕様書をコピーして貼り付けていることが多く、保守ではメンテナンスされない場合もあることから、付加価値を感じている人が少なかった。よって、プログラム設計を見直すことによりコスト削減できるのではないかと考えた。

2.改善したいこと

品質を保ちながらプログラム設計の無駄な作業を減らす。

3.改善策を導き出した経緯

当組織では独自の開発フレームワークを使っており基本的な画面の動きはパラメータを設定するだけで実現できる。実装が必要となるのはビジネスロジックのみであり、実装量は一般的なシステム開発に比べて少ないはずである。過去実績を確認したところ、全体の約 3 割はビジネスロジックが不要なプログラムで、約 3 割はコーディング量が 77Step 以下だった[1]。よって、全体の 6 割はプログラム仕様書が不要なほど簡単なはずで、プログラム仕様書が必要なプログラムは「ごく一部」ではないかと考えた。

そこで、開発予想規模からプログラム仕様書の要否を判断できるのではないかと考え検討したが、規模は小さくても一部の処理はプログラム仕様がないと作成するのが困難であると分かった。開発予想規模から要否を判断するのは断念したが、部分的には不要と判断できるのではないかと考えた。そこで仕様の 1 つ 1 つにプログラム仕様の要否を判断して、必要な箇所のみプログラム仕様を作成する改善策を導き出した。そうすれば、プログラム設計作業が減り、コスト削減に繋がると考えた。

4.改善策の内容

図 1 に外部仕様書のサンプルを示す。外部設計時に仕様の 1 つ 1 つにプログラム設計が必要かどうか判断し(図 1)、必要な個所のみプログラム設計することにした(図 2)。全ての仕様がプログラム設計不要となった場合プログラム設計は実施せず、プログラム開発に進む。そうすることで、機能によってはプログラム設計が不要となるもしくは設計する箇所が減る為、コスト削減に繋がる。

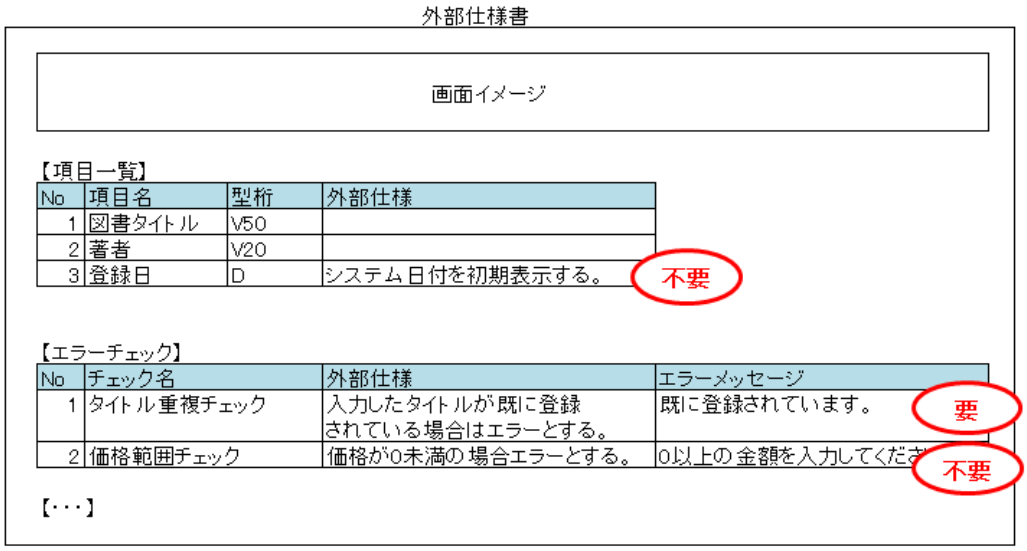


図 1　プログラム設計要否判定イメージ

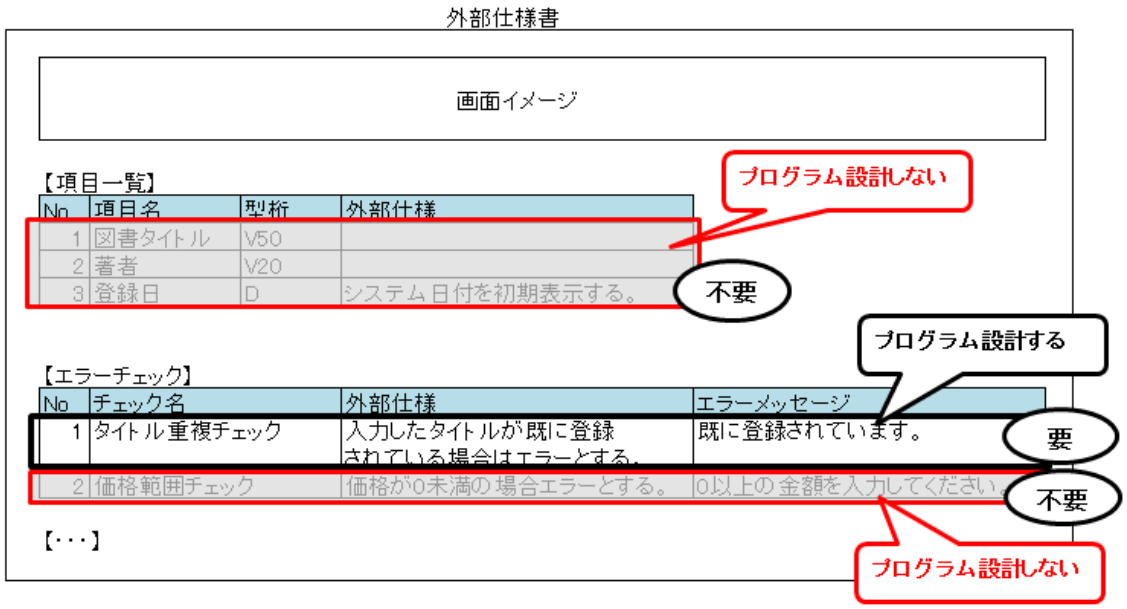


図 2　プログラム設計イメージ

5.改善策の実現方法

当組織では外部設計やプログラム設計で作成した設計情報をデータベースに保存し、その情報をそのまま利用して機能として動作する開発フレームワークを利用している [2]。今回の改善ではそのフレームワークの設計ツールに外部設計で登録した１つの仕様に対してプログラム設計の要否判定を入力する画面(①)や照会する画面(②)、入力した要否判定を見ながらプログラム設計を進められる機能(③)を追加した。加えて、設計する際に必要となる基準や手順などを開発標準として整備した。

(1)開発ツールの改善

① プログラム設計要否判定入力機能

外部設計時に手間なく要否判定の結果を登録できるように、1 つの仕様を記述する画面にプログラム設計要否を入力する欄を追加した。また、プログラム設計者に判定の意図を伝えられるように、判定した理由を入力する欄も追加した（図3)。

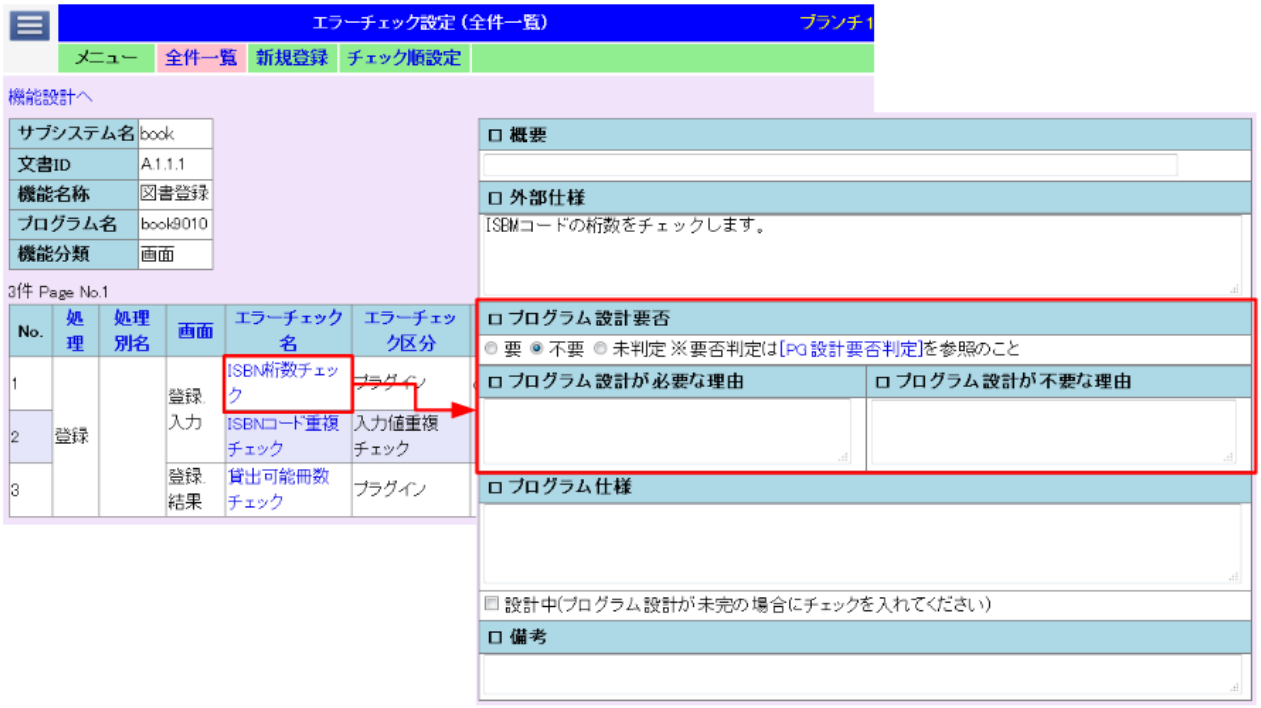


図 3 プログラム設計要否判定入力画面

② プログラム設計要否判定照会機能

この仕組みを導入するとプログラム設計以降の工数見積ではプログラム設計がどのくらい不要となったか把握する必要がある。その為、プログラム設計が不要となった仕様の件数や率をサブシステム単位、機能単位で確認できる画面を作成した。また、要否判定が漏れていないか、判定は正しいかを確認する為に仕様毎に要否判定結果を一覧で照会できる画面を作成した(図 4)。

図 4 プログラム要否判定照会画面

③ プログラム設計ナビゲート機能

プログラム設計要否判定結果を元に'要'の箇所のみプログラム設計を実施してみたところ、仕様の入力個所があちこちに点在し、効率面で問題があることが分かった。この問題を解消する為に、エラーチェック、更新処理などのプログラム設計項目毎に要否判定と設計の実施状況が分かる画面を作成した。プログラム設計要否が'要'でプログラム仕様が記載されていない個所をプログラム設計が必要な箇所と判断して'未着手'と表示した(図 5)。

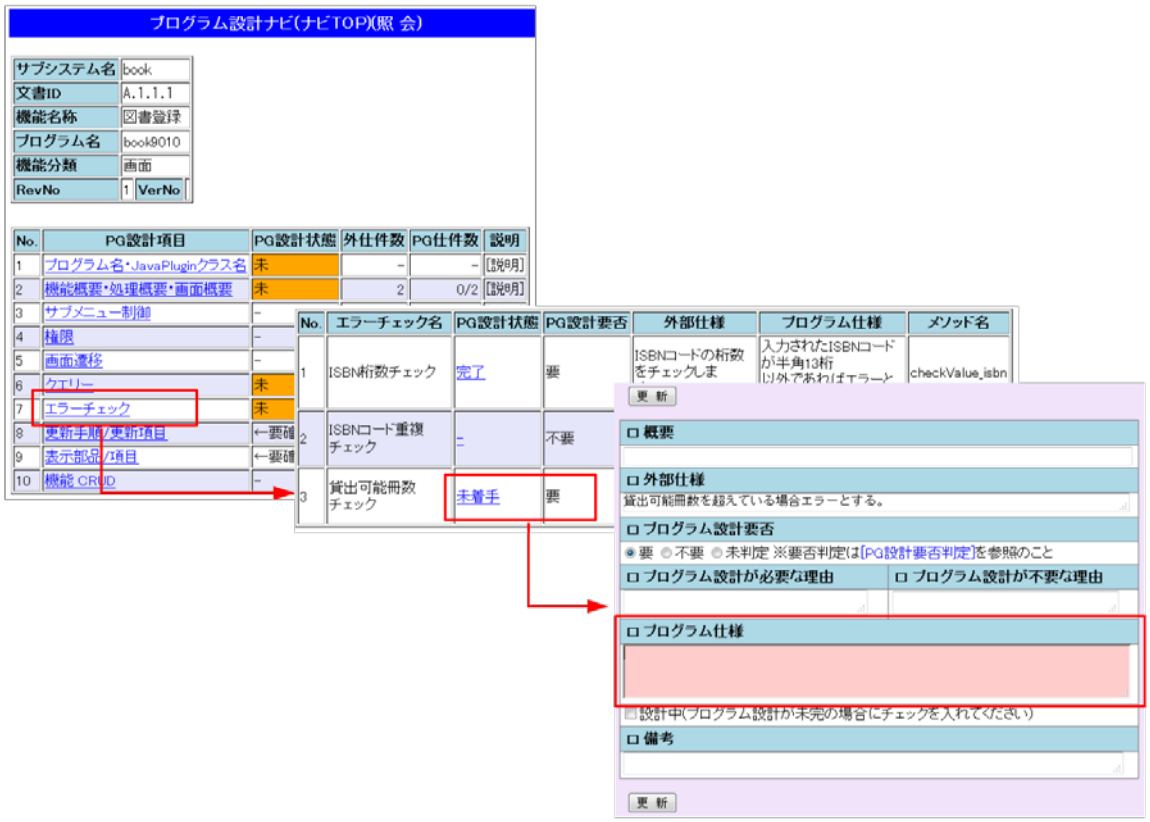


図 5 プログラム設計ナビゲート画面

(2)開発標準の整備

実施する人によって判定結果が変わらないように、プログラム設計要否を判定する為の基準を作成した(図 6)。また、誰でも容易に作業できるように、基準を使ってどのように判定を実施するかの手順(図 7)や、具体例を載せた判定例(図 8)も作成した。

プログラム設計要否判定基準

外部仕様書の記載箇所に対して以下を満たす場合はプログラム設計を不要とする。
1．仕様が論理的に特定されている
2．プログラム設計を必要とする仕様ではない

1．仕様が論理的に特定されている
判定対象の仕様が以下の全ての項目を満たしていること。
a．処理に使用するすべての値、及び取得方法が特定されていること
b．・・・
c．・・・
2．プログラム設計を必要とする仕様ではない
以下のケースに当てはまるものは1．を満たしていてもプログラム設計が必要とする。
a．実装方法により保守性(解析性、変更性、安定性、試験性)が悪化する可能性がある
b．・・・
c．・・・

図 6 プログラム設計要否判定基準

プログラム設計要否判定実施要項

手順1）処理に使用する全ての値、及び取得方法が特定されているか(基準1－a)を判定する。
手順2）データの加工、条件の判定が特定されているか(基準1－b)を判定する。
手順3）・・・
手順4）・・・
・・・

図 7 プログラム設計要否判定実施要項

プログラム設計要否の判定例

○外部仕様書の記載内容

No	チェック名	外部仕様	エラーメッセージ	表示場所
1	登録可能チェック	契約先C、利用実績年月が既に請求額確定している場合エラーとする。	すでに請求額確定している契約先の為、登録できません。	項目

基準1-a
使用する値は画面に表示されている為、
特定されている

基準1-b
確定を判断する条件がER図を見ても明確に分からない為、
特定できない

基準1-c
エラーメッセージの内容及び出力場所が明記されている為、
特定されている

⇒基準1－a、1－cは満たしているが、基準1－bを満たしていない為、仕様が論理的に特定できない。
よって、**プログラム設計要否は「要」**と判断する。

図 8 プログラム設計要否判定例

6.改善による変化や効果

プログラム設計要否判定の導入により、システム全体でどのくらいプログラム設計が不要となるのか、プログラム設計工数が削減できるのかを検証した。過去に開発した１システムの設計情報を使ってプログラム設計要否判定を実施し、削減率の計測と削減工数(予想)の算出を行い、評価した。

(1)プログラム設計不要率(削減率)

１システムの設計情報からプログラム設計要否を判定して、システム全体でプログラム設計が不要となる率を算出した。その

結果、システム全体で仕様の 60%についてプログラム設計を不要にできることが分かった。

機能数	要		不要		計
	件数	率	件数	率	件数
46	537	40%	822	**60%**	1,359

(2)削減工数(予想)

1 システムから 2 機能をピックアップして、プログラム設計要否判定と'要'と判定した仕様に対するプログラム設計を実施し、その作業時間を測定した。過去にプログラム設計した当時の工数と比較することで、プログラム設計の削減工数を算出した。その結果、外部設計及び PG 設計で、1 機能あたり 12～13%の工数削減が見込めることが分かった。

機能名	要		不要		計
	件数	率	件数	率	件数
機能A	1	5%	21	**95%**	22
機能B	2	8%	22	**92%**	24

(単位:MH)

機能名	過去実績		見込		削減見込
	外部設計	PG設計	外部設計	PG設計	
機能A	12	9	12.75 (+0.75)	5.85 (−3.15)	**2.75 (13%)**
機能B	20	6	20.25 (+0.25)	2.70 (−3.30)	**3.05 (12%)**

見込工数の算出方法について

・外部設計

過去実績の外部設計工数にプログラム設計要否判定を実施した工数分を足した値とした。

例) 機能 A 「過去実績の外部設計工数(12h)＋プログラム設計要否判定の工数(0.75)＝12.75h」

・PG 設計

要否判定が'要'となった仕様についてプログラム設計した時間を計測し、1 仕様あたりのプログラム設計工数を算出した。その工数に不要となった仕様数を掛けた分は削減できた工数と推測し、過去実績から引いた値とした。

例) 機能 A 「過去実績の PG 設計工数(9h)

－1 仕様あたりのプログラム設計工数(0.15h)×'不要'となった仕様数(21)＝5.85h」

7.改善活動の妥当性確認

過去システムでの検証結果からプログラム設計が不要となる仕様はシステム全体で 60%あり、必要な個所のみプログラム設計することで 1 機能あたり 12～13%の工数削減が見込めそうである。改善策を検討する際に過去実績から全体の 6 割はプログラム仕様書が不要なほど簡単だと推測した結果とも合っている。また、プログラム設計要否の判定結果をレビューすることで、判定間違いによるプログラム設計の漏れ等は防止できる為、品質は保てると考える。

よって、施策は妥当だと判断し、今後は実際の開発プロジェクトに適用し評価する。

A. 参考情報

[1] 中村 伸裕, “Software Product Line の実践 実装用ソフトウェア部品の開発と全社展開”, SPI Japan 2013, 2013

[2] 川口 晃史, “Software Product Line の実践 プログラム開発の効率化を目指した設計資産の構築”, SPI Japan 2013, 2013

1C4「製品ラインナップと開発制約に基づく丁度良い仕様の開発方法の提案」髙橋拓未（デンソー）

<タイトル>

製品ラインナップと開発制約に基づく丁度良い仕様の開発方法の提案

<サブタイトル>

－

<発表者>

氏名（ふりがな）：髙橋 拓未（たかはし たくみ）
所属：株式会社デンソー

<共同執筆者>

氏名（ふりがな）：林 健吾（はやし けんご）
所属：株式会社デンソー

<主張したい点>

既存製品にラインナップが追加され，ラインナップが崩れないよう差別化した仕様開発が必要となった．ブランドイメージを分析するための軸を活用して仕様の選択肢を立案し，プロトタイピングで検証した．その結果，開発期間やコストなどの開発制約を満たしながら，丁度良い仕様を確立することに成功した．

<キーワード>

要求仕様開発，プロセス設計，ポートフォリオ分析，スパイラルモデル，プロトタイピング

<想定する聴衆>

要求仕様開発業務に携わるエンジニア，複数の製品ラインナップの差別化を維持しながら要求仕様を立案する業務に携わるエンジニア，ソフトウェアエンジニア

＜活動時期＞

2016 年 12 月～ （継続中）

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
■ 改善活動を実施したが、結果はまだ明確ではない段階
□ 改善活動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，障害物をセンシングし衝突回避を支援するシステムの制御ソフトウェアを開発している．従来センサと従来 HMI 仕様を採用したシステム A に加えて，高性能センサと高機能 HMI 仕様を採用したシステム B を開発中で，これにより 2 ラインナップ化が進んでいる．ある車種において，当初はシステム B を採用する予定だったが，センサだけをシステム A のものに変更する要求を顧客から受けた．同時に，システム A の表示とブザー音をシステム B と揃えたい要求も顧客から受けた．しかし，開発にあたり制約があった．センサ性能差ゆえ，システム A の HMI 仕様をそのままシステム B の HMI 仕様に置き換えることが原理的に不可能という性能上の制約．また，システム B に仕様を近づけ過ぎると，ラインナップの差別化が図れないというラインナップ上の制約，また，システム B の開発を進めておりシステム A の開発を行うには開発力が不足しているというリソース上の制約，そして，システム B にできるだけ揃えようとすると納期を満たせないという納期上の制約があった．このような状況の中で，制約上できる範囲内で，ラインナップとして差別化ができつつ，比較的高級になったと感じられるリーズナブルな”丁度良いシステム要求仕様”の提案・確立が必要であった．発表者は，システム要求仕様を提案・確立するため，活動を開始した．

2.改善したいこと

顧客と要件交渉を重ねたが，実現可能な仕様の折り合いがつかないという問題が発生し，システム要求仕様が定まらずに時間が過ぎていく状況が発生した．その理由は，安価にしたいが製品として見劣りさせたくないという要望と，性能上・開発上の制約との衝突が発生しているためであった．原因は，発表者が要望や制約を満たす適切なシステム要求仕様を提案できていないことにあった．従って，適切かつ妥当なシステム要求仕様を立案・確立するプロセスの定義が課題となった．

3.改善策を導き出した経緯

システム要求仕様を立案・確立するプロセスを決める上で参考になりそうな関連研究を調査した．

・要求を要求仕様化するプロセス…顧客から獲得した要求を分析し，実現可能な要求に落とし込むため要求交渉を繰り返し行いながら，要求仕様を定義するプロセスが提案されている[1]

・スパイラルモデル…開発の各フェーズで生じ得るリスクを抽出し，プロトタイピングによる検証を通じてリスクを低減しながら，要求仕様や設計仕様を確立するモデルが提案されている[2]

・ポートフォリオ分析…市場における製品のシェアと成長率のように，２軸を掛け合わせて得られる領域のどのポジションに資源を振り分けると有効であるかを分析する方法が提案されている[3]

これらの関連研究から，顧客から獲得した要求を分割し，一つ一つに対して顧客と要求交渉を重ねる必要があること，顧客と要求仕様の合意を得る上で，必要に応じて適した方法でのプロトタイピングを行うと効果的であること．また，交渉材料としてのシステム要求仕様の選択肢を得るためには，選択肢を作成するために最適な軸を選択することが必要であることに着眼した．

4.改善策の内容

システム要求仕様を立案・確立するプロセスを以下の通りに定義した．前提として，顧客からは何らかの手段によって要求を獲得できているとする．

１．要求分析に入る準備

１－１．要求仕様の選択肢を作成するための軸を選択する．
軸の個数に制約は無い．

２．要求分析と選択肢の立案

２－１．要求を分割する．

２－２．分割した要求１件１件に対して，軸を使って要求仕様の選択肢を立案する．
選択肢毎の技術的課題の分析と開発コスト見積もりを行う．必要に応じてプロトタイピングを行う．

（開発チーム内部のプロトタイピング）

２－３．要求交渉の材料として，要求仕様の選択肢を組み合わせた仕様セットを数パターン用意する．

２－４．パターン毎のトータル工数を見積もる．

３．要求交渉

顧客と要求交渉をして，顧客が納得できる要求仕様のセットを合意する．

必要に応じて要求仕様のプロトタイピングを行い提示する．（開発チーム外部のプロトタイピング）

４．要求交渉の中で顧客から新しい要求を獲得したら，２．～3．を繰り返す

5.改善策の実現方法

改善策で定義したプロセスで重要となる軸の選択について下記の通り説明する．

① 軸の選択

要求仕様の選択肢を作成するための軸として，「コスト」，「期間」，「センサ性能」に加えて「Quality」，「Emotion」の軸を選択した．「Quality」と「Emotion」は，ブランドイメージのポジションを分析するための軸として定義されている（図１）[4]．この軸を掛け合わせることで「Premium」，「Respect」，「Love」，「Ignorance」の４つのポジションが定義されている．システム B は，このポジションでは「Premium」に位置する．従って，システム A で差別化を図るには，「Respect」か「Love」のポジションを狙うべきである．品質は落とせないことから，ここでは「Respect」を選択した．

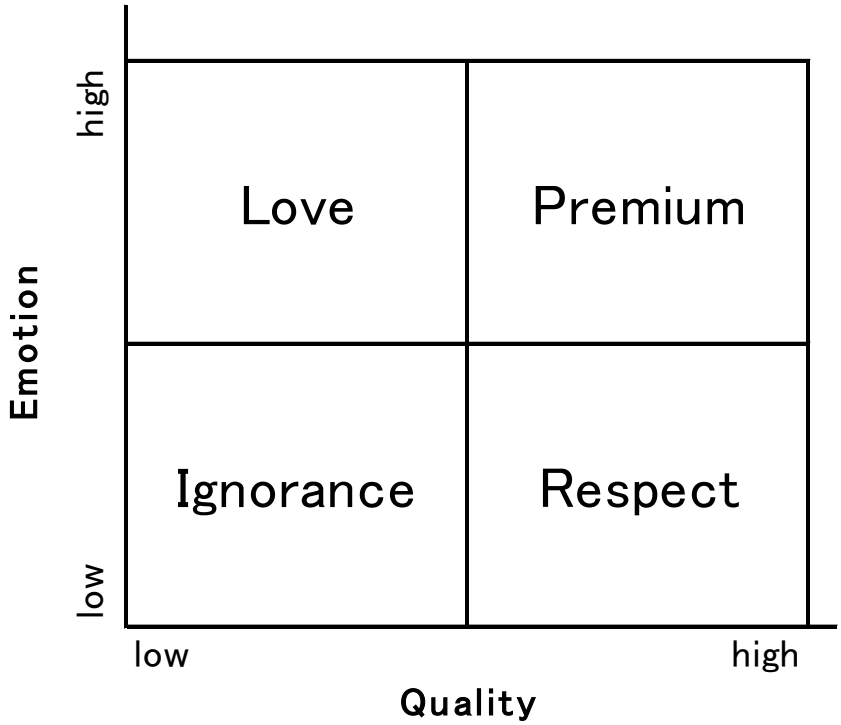


図1 ブランドイメージのポジションを分析するための軸

② 要求仕様の選択肢の立案

「Premium」を「Respect」へ落とす軸として「Emotion」がある．従って，システム A を「Respect」に位置付けるためには，「Emotion」を落とせばよい．ここでは，この「Emotion」を「高級感」と定義した．そして，要求仕様の選択肢を，システム B に対して高級感を落とした仕様として立案した．

③ 要求交渉

顧客側のステークホルダーとして，開発担当者，開発管理者（マネジャー）のキーパーソンを把握した．要求交渉を繰り返しながら，適切なタイミングでキーパーソンに対し交渉の場への参加を依頼した．官能評価が重要な要求仕様は，実機による

プロトタイプを提示し，顧客協力の元，実車での官能性評価を実施した.

6.改善による変化や効果

① 改善による効果

要求交渉を重ねることによって，当初は発見できなかった要求を発見することができた．また，適切かつ妥当なシステム要求仕様の選択肢を作りだすことができ，仕様確立の目処付けができた.

② 改善による変化

要求交渉は，システム要求仕様の選択肢の共有までは円滑に行えたが，仕様を合意することに時間を要した．開発コスト，売価，機能差別化を明示することで，双方にメリットがあることを伝えた．かつ，プロトタイプを準備することで，顧客協力のもと，実車両で官能性の評価を実施でき，提案仕様の妥当性を確認できた．もし評価が実施できなければ，仕様セットの合意が遅れ，納期遅延のリスクが高まっていた.

7.改善活動の妥当性確認

立案したシステム要求仕様の選択肢を，プロトタイプとして提示できたことで，要求と制約にミートするシステム要求仕様を導き出すことができた．今後は本活動をプロセスとして資産化し，製品ラインナップが追加された際に，要求仕様の選択肢を立案し顧客と合意を達成するプロセスとして活用していく.

A. 参考情報

[1]飯村結香子・斎藤忍・青山幹雄，REBOK に基づく要求分析実践ガイド，NTT ソフトウェアイノベーションセンタ，株式会社近代科学社，2015

[2]Barry W. Boehm, A Spiral Model of Software Development and Enhancement, IEEE, 1988

[3]プロダクト・ポートフォリオ・マネジメント，https://ja.wikipedia.org/

[4]PHILIPP G. ROSENGARTEN & CHRISTOPH B. STUERMER, PREMIUM POWER, Palgrave Macmillan; 2005

1D1「CMMI を軸としたニアショア開発における品質改善活動」井上靖章（クオリサイトテクノロジーズ）

<タイトル>

CMMI を軸としたニアショア開発における品質改善活動

<サブタイトル>

<発表者>

氏名（ふりがな）：井上　靖章（いのうえ　やすあき）
所属：　クオリサイトテクノロジーズ株式会社　品質管理部

<共同執筆者>

氏名（ふりがな）：
所属：

<主張したい点>

改善活動に取り組む前の当社は、地方（沖縄）にあることもあり、日本語が通じる価格が安いオフショア会社という程度の位置づけであった。状況を打破するため、ほぼ何もない状態から CMMI のモデルを参考に組織としての品質改善活動に着手した。その結果、CMMI レベル 3 達成、レベル 4 を達成し組織が成熟したことにより、品質の向上、対外的な評価向上、業績向上と効果のある取り組みが出来た。

<キーワード>

地方、CMMI、失敗プロジェクト率の減少

<想定する聴衆>

地方 IT 企業、これからプロセス改善を実施しようとしている IT 企業および組織、プロセス改善活動で悩んでいる IT 企業および組織

＜活動時期＞

2012 年～2017 年（継続中）

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（　　）

<発表内容>

1.背景

CMMI を軸とした品質改善活動開始前、当社では成功プロジェクトと失敗プロジェクトがはっきり分かれていた。

若い会社だったこともあり、組織としての品質保証や標準プロセスなどが整備されておらず、経験を積んだ PM 配下のプロジェクトは成功し、未経験および経験が浅い PM 配下のプロジェクトは失敗する傾向があった。つまり、当社のプロジェクトの成否がヒトに依存している状況であったことを示していた。そのため、組織として品質改善活動を行い、品質保証、標準プロセスを整備することで、プロジェクトの失敗確率を減少させる施策を検討した。

また、当時は、地方（ニアショア）という立地から価格面でのメリットが受注理由の大半を占めていた。そのため、将来、価格競争に巻き込まれる懸念があり、その対策を講じる必要に迫られていた。

2.改善したいこと

改善の目的は、失敗プロジェクトの減少と高い品質の維持である。これらを達成することを通じて低価格のニアショア会社から、プロジェクト管理と品質管理に優れ信頼できる会社へポジションを変えることを図りたかった。また当社の業務の実情から、新規開発以外の保守開発プロジェクトにも適用できる独自のプロセスを組織のノウハウとして持ちたいと考えた。

3.改善策を導き出した経緯

ヒトに依存している状況を解決するためには、成功プロジェクトのノウハウを蓄積して展開し、同種の問題を是正することが重要と考えた。そのため、品質管理部主導で、組織の仕組みとしてノウハウの蓄積と展開するための方法を検討した。当時、沖縄県では、CMMI を推進する動きもあり、CMMI の考え方を適用し CMMI に沿って組織の成熟度を高めていくことで、問題を解決できるという結論に至った。

また価格競争に巻き込まれた際の解決策として、CMMI を用いた改善活動を継続して高い品質を維持していくことで、他社との差別化を図ることが出来ると考えた。

4.改善策の内容

ヒトに依存したプロジェクト運営をなくすため、組織としての品質保証と標準プロセスを作成した。成功プロジェクトのノウハウと CMMI のモデルを参考にしつつ、プロジェクトマネージャにも参画してもらった。プロジェクトマネージャに参画してもらった理由は、実業務を理解している人間が携わることで現場の意見が反映された標準プロセスとなることを狙った。多忙なプロジェクトマネージャの協力が得られたのは、組織が抱えている問題について、プロジェクトマネージャも会社全体で改善する課題だと共通認識を持っていたためである。

また、既存プロジェクトに対し、標準プロセスの適合度合いをテーラリング結果として可視化した。その上で、プロジェクト側の課題の原因をプロジェクトマネージャと品質管理部で分析した結果、原因が標準プロセスと乖離したプロジェクト独自のプロセスにあった場合、標準プロセスを取り入れることを提案し、適用してもらった。標準プロセスの作成経緯を説明し、適用可否については、プロジェクトと合意する形で進めた結果、適用することに反対するプロジェクトや消極的なプロジェクトが出ることもなく標準プロセスを展開できた。

5.改善策の実現方法

成功プロジェクトと CMMI のモデルを参考にした標準プロセスを作成し、既存プロジェクト、特に失敗しているプロジェクトに対して GAP 分析を実施した。そして、分析結果を基に、成功したプロジェクトのプロセスをテーラリングし既存プロセスに適用してもらった。

成功事例の展開となるため、既存プロジェクトにとっては、標準プロセスとの GAP の解消とプロセス改善を同時に進ませることとなった。さらに、当社プロジェクトの特色として、中長期にわたる保守開発プロジェクトが多い点がプロジェクトへの展開をスムーズに行うことに寄与した。それは、長いプロジェクトであっても、4 半期計画と年間計画を立て、そのそれぞれで PDCA サイクルが存在したこと、及び顧客を含めプロジェクトも改善活動に積極的であったことである。
具体的な展開方法として、事業部長と行う月次の監査の場を活用した。各プロジェクトの課題は、月次で実施される事業部長と品質管理部の監査にて共有され、対応策について協議される。この監査の場で、標準プロセスを取り入れるかの判断が行われ、詳細な適用方法などは、品質管理部が現場のプロセスを理解した上でアドバイスという形で提案され、プロジェクトと合意したのち適用される。
CMMI の観点では、以下、２つのプロセス領域に注力した。１つは、プロジェクト進行中に事業部との認識齟齬を是正するための PP（プロジェクト計画策定）。もう１つは、プロジェクトを失敗させないため、リスクの予防、と早期に是正するための RSKM（リスク管理）に注力した。特に RSKM は、リスク発生時の影響とリスク顕在化時にかかる費用として数値化することで対策、予防措置と是正措置を講じるタイミングについて組織側からも指示できるようになり失敗の未然防止につなげることが出来た。

6.改善による変化や効果
既存プロジェクトに適用したところ、失敗していたプロジェクトの立て直しが出来るようになってきた。
１つ目の理由は、プロセスのドキュメント化が進んだことにより属人化していた作業が少なくなり、手順不備などの減少、人依存による品質差異が極小化したこと。２つ目の理由は、「5.改善策の実現方法」にも記載したが、RSKM（リスク管理）により客観的に予防措置、是正措置を講じることが出来るようになったことが効果に繋がったと分析している。
結果として、改善取組前と比較し失敗プロジェクト率の減少という効果があった。
また品質が高まってきたことで、高い品質が求められる金融系システム開発の受注量増加と、顧客満足度が向上した。
当社は、長期間におよぶエンハンスプロジェクトが多いため、新規プロジェクトに適用した効果については、今後、検証を行っていく。
また、CMMI レベル４を達成した効果についても、２０１６年年末に達成したため検証中となっている。

7.改善活動の妥当性確認
CMMI により、滞りがちであった PDCA サイクルが制度化されたことで品質改善活動のサイクルが定常化された。
CMMI の成熟度がレベル３からレベル４に上がったことで、品質改善活動に対するプロジェクト側の興味とその展開依頼が高まった。
顧客意識としても、「ニアショアだから単純に価格が安い」から、「ニアショアだけれど品質レベルを加味すると価格が安い」に変化してきている点が感じられる。
上記、理由から、本改善を継続させ最終的に、CMMI レベル５を達成したいと考えている。

1D2「IT サービスマネジメントの観点を考慮した運用保守プロセスの評価と改善」中村英恵（NTT データ)

<タイトル>

IT サービスマネジメントの観点を考慮した運用保守プロセスの評価と改善

<サブタイトル>

サービスのための CMMI レベル 3 達成の取組

<発表者>

氏名（ふりがな）：中村英恵（なかむら　はなえ）
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）：花田賢太郎（はなだ　けんたろう）
所属： NTT データ

<主張したい点>

システムの運用保守を主体としたサービス提供について、プロセス改善モデル「サービスのための CMMI(CMMI-SVC)」を活用した改善活動を実施し、改善の結果を CMMI 公式評定で確認した。日本国内初となる CMMI-SVC のレベル 3 達成について、その改善内容を振り返る。

<キーワード>

サービスのための CMMI、サービス提供、運用保守

<想定する聴衆>

サービス提供プロセスの改善に興味がある人
開発のための CMMI の経験があり、サービスにも CMMI を検討している人

＜活動時期>

2016.5-現在継続中

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動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 データでは、システム開発後の運用保守（サービス提供）も請け負うことが多く、ビジネスに占める割合も高い。
運用保守プロジェクトでは、これまで現場の IT サービスマネージャが独自にプロセス改善活動をおこなってきたが、プロセス改善活動を活性化させるため、目標設定やメンバの動機づけを工夫したいというニーズがあった。また、プロセスの充実度について世間一般で通用する視点を取り入れてはどうかという意見も出ていた。
そこで、システムが所属する組織のメンバと、コーポレートの SEPG が協同し、プロセス改善モデルであるサービスのための CMMI(CMMI-SVC)を用いた改善活動を行うことになった。

2.改善したいこと

組織の運用保守プロセスが世の中の基準(CMMI-SVC)と照らしてどの程度かを知ること
その結果として、運用保守プロジェクトに必要なプロセス一式を整備すること

3.改善策を導き出した経緯

NTT データでは開発プロジェクトの改善を行う組織に対しては、これまで開発のための CMMI(CMMI-DEV)に基づくプロセス改善を推進してきた。
CMMI-SVC は CMMI-DEV と共通の枠組みで作られているプロセスモデルであり、すでに CMMI-DEV の改善経験がある組織にとっては理解、実装ともしやすいというメリットがある。(参考文献[1])
また、全社 QMS の枠組みでは、IT サービスマネジメントの改善活動を推進しており、その中でもモデルベースの改善を進めていくことを手段の一つとして掲げている。

4.改善策の内容

CMMI の改善サイクルに従い、立ち上げ、ギャップ分析、アクションプラン実施、公式評定の各フェーズを定義した。
当社では以前から CMMI の改善活動向けに開発した「改善活動 Startup の」コンテンツを使用しているが、CMMI-SVC については初回だったこともあり、プロセス領域トレーニング教材やギャップ分析のワークシートなどの拡充を行った。(参考文献[2])

5.改善策の実現方法

＊CMMI-SVC モデルの理解と実践について

CMMI-SVC は「サービス業全般」向けの改善モデルであり、IT システムの保守運用に特化しているわけではない。
そこで個々のプロセス領域の評価分析、実装においては ITIL の知識体系を応用することにした。
立ち上げフェーズにおける、プロセス領域のトレーニングでは、教材に ITIL の該当するプロセスを例示したり、ギャップ分析やアクションプラン実施フェーズにおいてはコーポレートの ITIL 上級資格保持者、現場の有識者を交えた議論を行うなど、プロセス実装の How の部分にできるだけ明確な基準を示せるようにした。
また、サービス提供においては、その要求品質は一様ではなく、SLA などシステムの契約ごとに決まるという性質が強い。たとえば、システム運転時間は平日日中なのか、24 時間なのか、といったことである。そこでギャップ分析と評定の各フェーズでは、SVC 独自プロセス領域について、プロセス実装は SLA の置かれた状況と対比して十分か、という議論を行った。

＊プロセス文書の実装について

アクションプラン実施フェーズにおけるプロセス文書の整備では、すでに適用していた開発のための CMMI と文書体系を同一にし、CMMI-DEV と CMMI-SVC の両方にあるコアプロセス領域は既存のプロセス文書の修正、CMMI-SVC 独自プロセス領域は既存の文書を修正のうえ体系に追加、と、全体の質を担保しつつ対応工数を抑えることができた。

6.改善による変化や効果

当社組織の運用保守プロセスが世の中の基準と照らしてどの程度かを知ること

→ギャップ分析と評定により、客観的な視点で評価することができた

初めてのギャップ分析であったにも関わらず、強みが多く抽出された。

弱みは、プロセスの制度化、コンティンジェンシープランの妥当性確認、リスク分析などであった。

運用保守プロジェクトに必要なプロセス一式を整備すること

→アクションプラン実施により、CMMI-SVC 成熟度レベル 3 に沿った整備ができた

7.改善活動の妥当性確認

サービス提供におけるプロセス改善活動に弾みをつけること

→プロセス改善上の目標を設定し、役割を明確にすることで組織的に動けるようになった。

・プロセス改善目標に則したリスク分類や故障回復演習の企画を組織的に検討することができるようになった。

・組織外のメンバが参加することで、モデルの要求事項以上に実施できている点も多く発見でき、推進にメリハリがつくようになった。

・運用保全に脚光があたったことで、普段自分たちがやっている仕事と改善の価値を再確認できた。

コーポレートの視点

→得られたノウハウを全社 QMS 活動にフィードバックし、グループ内で共有できるようになった。

国内初 CMMI-SVC レベル 3 達成のニュースリリースに伴い、事例情報提供や支援説明依頼の問い合わせが多く寄せられており、グループ内での関心は高い。

A. 参考情報

[1]CMMI for Services V1.3(CMMI Institute)

[2]SJ2016 の中村発表：プロセス改善活動をスムーズに立ち上げるための取り組み <その 2>

1D3「CMMI を活用した社内版プロセス評価モデル構築による全社品質・生産性向上への貢献」柳田礼子（NEC)

<タイトル>

CMMI を活用した社内版プロセス評価モデル構築による全社品質・生産性向上への貢献

<サブタイトル>

<発表者>

氏名（ふりがな）：柳田　礼子（やなぎだ　れいこ）
所属：　日本電気（株）

<主張したい点>

CMMI の要求事項は初心者には理解しづらく、アセスメントはコストがかかる一方で、現場は効果を実感しづらい。これを解決するため、データ分析によりプロセス成熟度向上の効果を実証し、現場の改善への意識を向上させたうえで、有識者でなくても容易に、正確にプロセスを評価できる CMMI を社内版に改良したアセスメントモデルを構築して全社展開した。

<キーワード>

プロセス改善、データ分析、CMMI、アセスメント、組織的改善、成熟度レベル、能力度レベル、実績ベースライン

<想定する聴衆>

SEPG

＜活動時期＞

2016 年 4 月～継続中

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
□ 改善活動の結果が明確になっている段階
■ その他（一定の成果（結果）は明確になったが更なる改良に取り組み中）

<発表内容>

1.背景

自身は、品質・生産性を向上することを目的として、社内の各事業体が自律的に組織的改善サイクルを回すことができるよう、全社的な改善活動推進を担務している。

組織が自律的に組織的な改善活動を実施するためには、事実に基づき、現状の課題を正しく認識する必要がある。プロセス実施状況を正確に診断して課題を抽出するために、CMMI は有効なツールとなり得る。当社では、2011 年度～2012 年度に CMMI 推進活動をタスクフォース化して各事業体の成熟度の底上げを図った。社内のソフトウェア開発全組織を、複数のリードアプレイザ有資格者が分担して簡易アセスメントして現状レベルを診断し、各組織にて改善した結果を約 1 年後に再アセスメントした。全体的な成熟度レベルは底上げしたが、アセッサの継続的なリソース確保が困難なこともあり、一部の組織を除いて一過性の改善活性化に留まった。

2.改善したいこと

一般的に、CMMI の要求事項は初心者には理解しづらく、アセスメントはコストがかかる。また、組織の成熟度を向上するための改善活動については、顧客のクレームや GP 率の悪化などの顕在化した問題を解決する改善活動に比べて必要性の認識が低い。改善効果について現場要員の納得感が得られにくく、活動の動機付けが難しい場合が多い。

当社でも、組織的な改善サイクルの確立、および、全社的なプロセス改善状況の見える化に対する幹部のニーズはあるものの、現場組織では、外部要因がある場合を除いて、CMMI が積極的に活用されている状況にはない。CMMI を用いたアセスメントに自発的に取り組む組織は少なく、プロセス改善活動にモデルが有効活用できている組織は限定的である。

3.改善策を導き出した経緯

① CMMI 推進タスクフォースにおいて評価した結果を活用して、全社的に収集した品質・生産性に関係するプロジェクトのプロセスデータを成熟度レベル別に層別して分析した。結果として、CMMI の成熟度レベルを向上させることが最終的な品質向上に結び付くことを実証し、これにより、プロセス成熟度を向上させることの効果を示すことができた。

② 全社のプロセス実績ベースラインに対する各組織の立ち位置を分析してフィードバックし、さらに、分析手法の定型化や教育展開により、各組織にて一定の分析ができる環境を整備した。これにより、事実に基づく現状把握・課題抽出の必要性の認識が高まった。

③ データ分析に加えて、プロセスの評価結果も併せて現状認識することにより、組織の実態をより具体的に分析・考察することが可能になる。プロセス成熟度向上の効果、事実に基づく現状把握の必要性が認識された状況において、プロセス評価の仕組みを導入することが効果的と考え、CMMI を活用した当社オリジナルの評価モデルを提案した。

4.改善策の内容

各組織が自律的に継続的に改善サイクルをまわすために、導入するプロセス評価の仕組みは、有識者でなくても容易に正確に評価できる仕組みが必要であると考え、当社版オリジナルのアセスメントモデルを構築・展開した。

汎用的なプラクティスの記述を当社として実施すべき事項として具体化し、実施すべき事項にプラクティスを複数対応付けることにより、確認すべき事項を集約した。さらに、プラクティス実装（プロセス実践）の結果として、どういう状態であるべきか、という視点を確認観点として追加した。

5.改善策の実現方法

① 当社として実施すべきことが記述されている標準・規程・ガイドラインを基に、ソフトウェア開発・管理およびプロセス管理上の重要事項を抽出し、能力度レベル 1 を評価する項目と位置づけた。また、これらの実施によって期待される状態について、確認項目として抽出し、能力度レベル 2 および 3 を評価する項目と位置づけてモデルを構築した。

なお、現場での導入をスムースにするため、重点的に全社施策を展開している活動に絞って、対象とするプロセス領域を 8 つ

に限定したモデルを構築した。

② ３組織を試行対象とし、セルフチェックによるアセスメントを実施した。セルフチェックによる評価結果を、リードアプレイザがヒアリング及び文書レビューを実施して、結果の信頼性について検証した。同時に、試行組織より、セルフチェックにおける意見や感想を収集してモデルを修正・改善した。

③ 試行結果を反映したアセスメントモデルをもとに、１組織で再度アセスメントを実施した。結果に問題ないことを確認し、全社展開して約 30 の事業体にてアセスメントを実施した。

6.改善による変化や効果

✓ スモールスタートで現場の理解を得ることを方針として、アセスメントは強制力のない形で全社展開したが、展開した全組織にてアセスメントが実施された。これは、数年に亘って事実に基づく改善活動の基盤が構築された成果であると考える。

✓ コスト面では、当社内のリードアプレイザによる簡易アセスメントでは、標準的には、調査工数が約 1.5 人日/PJ であることに対し、今回アセスメントは約１人 H/PJ で実施できた。これは、社内の共通言を用いた確認項目の具体化により、CMMI 知識のない現場要員でも容易に評価できたことの効果である。

✓ 評価結果の信頼性については、プロセス実践の結果も観点に含まれていることにより、成果を伴わない過度なプロセスの実装や、セルフチェックでありがちな実態以上に高い評価値とすることを防ぎ、正確な評価結果が得られたと考える。実際に、全組織のアセスメント結果を分析したところ、相対的に結果指標の数値が悪い組織は、良い組織に比べてプラクティスの実装率が低いことが確認された。

✓ 今回は全組織とアセスメント結果について対面でレビューし、結果の読み解き方、改善策検討への活用方法を指導した。リードアプレイザによる簡易アセスメントではやや一方向の結果報告会であったことに比べると、今回は組織自身によるアセスメントのため、双方向の議論となり、より有益なフィードバックとなったことが副次効果として挙げられる。

7.改善活動の妥当性確認

今回アセスメントで明確になった課題を改善するための施策を立案する組織、プロセス成熟度向上することを目標として設定する組織が増え、自律的な改善活動への一助となったと考える。

また、アセスメント結果と組織の課題認識とが合致したことによる現場の納得感と、全組織の評価結果や抽出された課題に対する幹部層の納得感が得られ、本アセスメントの本格適用に繋がった。プロセス領域を８⇒１８に拡大して CMMI の成熟度レベル ３ までを網羅したモデルを構築すること、および、本アセスメントを全ソフトウェア開発組織で適用して定期的にアセスメントすることが決定した。

すでに全社課題についても改善施策を実行中であり、今後は定期的なアセスメントによりプロセス成熟度の向上度合いを見える化するとともに、アセスメント結果に基づく更なる改善を図り、プロセス実績ベースラインの改善に繋げる所存である。

1D4「パーソナルソフトウェアプロセスに基づく実践的ソフトウェアエンジニアリング教育の効果と自己改善活動におけるインストラクタの役割」梅田政信 (九州工業大学)

<タイトル>

パーソナルソフトウェアプロセスに基づく実践的ソフトウェアエンジニアリング教育の効果と自己改善活動におけるインストラクタの役割

<サブタイトル>

<発表者>

氏名（ふりがな）：梅田　政信（うめだ　まさのぶ）
所属：　九州工業大学　大学院　情報工学研究院

<共同執筆者>

氏名（ふりがな）：　片峯　恵一（かたみね　けいいち）
所属：九州工業大学　大学院　情報工学研究院
氏名（ふりがな）：　荒木　俊輔（あらき　しゅんすけ）
所属：九州工業大学　大学院　情報工学研究院
氏名（ふりがな）：　橋本　正明（はしもと　まさあき）
所属：九州工業大学名誉教授

<主張したい点>

九州工業大学は、パーソナルソフトウェアプロセス(PSP)とチームソフトウェアプロセス(TSP)を大学院教育に取り入れ、実践的なソフトウェアエンジニアリング教育に取り組んでいる。本発表では、2007 年度より実施している PSP コースの結果に基づいて、大学院教育においても社会人と同等の自己改善効果が得られ、高品質なソフトウェア開発に必要な知識とスキルの修得に PSP が有用であることを示す。また、PSP による自己改善には、講義と演習だけでなく、演習結果のレポートのレビューを通じた PSP インストラクタの指導が重要な役割を担っていることを示す。

<キーワード>

パーソナルソフトウェアプロセス，PSP、自己改善，PSP インストラクタ

<想定する聴衆>

ソフトウェアエンジニア，プロジェクトマネジャ，プロセス改善推進者，PSP インストラクタ

＜活動時期＞

2007 年―2016 年

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（　　）

<発表内容>

1．背景

情報システムに対する要求の高度化と複雑化に伴い、ソフトウェアの品質がビジネスの成功を決定づける大きな要因となりつつある。ソフトウェアの開発は、ソフトウェア技術者が自身の開発プロセスを自律的に管理せざるを得ない知識集約型の労働であるため、ソフトウェア技術者には、高品質なソフトウェア開発に必要な知識とスキルを修得することが求められている。

そこで、九州工業大学は、カーネギーメロン大学ソフトウェアエンジニアリング研究所(CMU/SEI)と連携し、パーソナルソフトウェアプロセス（PSP）[1]とチームソフトウェアプロセス（TSP）[2]を大学院教育に取り入れ、実践的な知識とスキルとを修得したソフトウェア技術者の養成に取り組んできた[3、4]。

2．PSP コースの概要と成果

九州工業大学における PSP コースは、SEI 認定の PSP トレーニングコースである PSP for Engineers と同等の内容を、主に博士前期課程 1 年生向けに実施する演習科目である。PSP for Engineers は、PSP-Planning と PSP-Quality をそれぞれ 5 日間で実施するのに対し、PSP コースは、各々を一つのクオータ期間に実施している。PSP-Planning ではプロセス規律、測定、見積り、および計画立案と追跡を、また PSP-Quality では品質マネジメントと設計標準について学ぶ。

同コースは、2007 年度から開始し、2016 年度までの 10 年間に累計 95 名が受講している。図 1 は、これらの受講者におけるテスト欠陥密度（単体テストにおいて発見修正された 1000 行当たりの欠陥数）の最大値、平均、最小値の推移を表したものである。横軸は PSP コースで課される演習課題の課題番号、縦軸は欠陥密度を表す。この結果から、課題の進捗につれて欠陥密度が減少傾向を示し、課題 8 における欠陥密度の最大値は、課題 1 における平均値とほぼ代わり無いことが分かる。

一方、図 2 は、生産性（単位時間当たりの開発規模）の推移を示している。この結果から、課題の進捗につれてプロセスに自己レビューが追加されるなど、プロセスが次第に複雑になるにもかかわらず、生産性は大きく変化しないことが分かる。これらのことから、生産性を低下させることなく、テスト欠陥密度が低い高品質なソフトウェアを開発できるスキルを修得できていることが分かる。これは、SEI による主に社会人向けの結果とほぼ同様である。

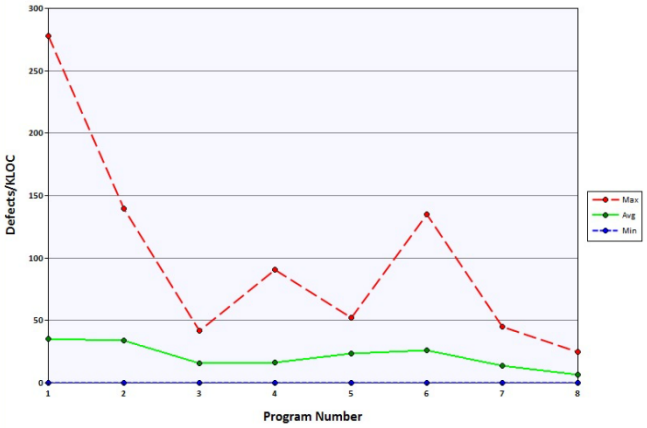


図 1　テスト欠陥密度の推移

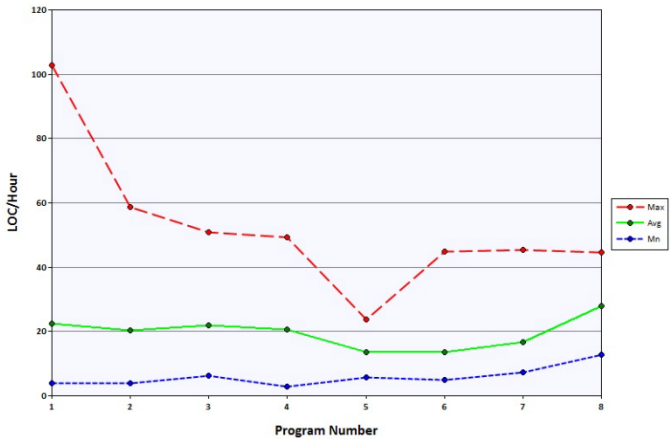


図 2　生産性の推移

3.　PSP インストラクタによる指導の効果

PSP コースは、講義と演習とから構成され、PSP インストラクタによる講義の後に、数 100 行程度の小規模なソフトウェアを開発する演習を行う。演習においては、受講者は、ソフトウェアの規模、分単位のフェーズ別時間、および欠陥の埋込と除去のフェーズ、欠陥種別、欠陥内容等を記録し、他の成果物と共に PSP インストラクタにレポートとして提出する。PSP インストラクタは、そのレポートを精査し、時間記録の漏れや誤り、欠陥の漏れや欠陥種別の誤り等を確認し、必要に応じて個別に指導する。

図 3 は、2013 年度～2016 年度の受講生の中で、毎レビュー時にプロセスデータの提供を受けた 26 名について、レポート不備が原因によるレポート再提出回数の割合を演習課題毎に示したものである。この結果から、欠陥種別の誤り等の不備が当初多いものの、課題の進捗につれて次第にスキルが定着し、結果としてレポート提出回数が減少していることが分かる。

一方、図 4 は、インストラクタの指導により欠陥種別の選択誤りが判明し、修正した割合の推移を示している。この結果から、当初、欠陥の約 4 割において欠陥種別を誤って選択しているものの、適切な欠陥種別を選択するスキルが次第に定着していることが分かる。

欠陥種別や欠陥の埋込と除去のフェーズ等は、どのフェーズにおいてどのように欠陥の埋込を防止し、どのフェーズにおいてどのように欠陥を除去するかに関わるため、プロセスの自己改善には極めて重要である。しかし、欠陥種別や埋込フェーズ、除去のフェーズの選択誤りは、受講者が自ら気付くことは極めて難しく、PSP インストラクタによる指導抜きでは改善が困難である。このため、市販の PSP 関連図書や SEI の WEB サイトよりダウンロード可能な文書等を用いて PSP を自学自習する場合は、必ずしも十分な改善効果を見込めない恐れがあることに注意する必要がある。

図 3　レポート提出回数の推移

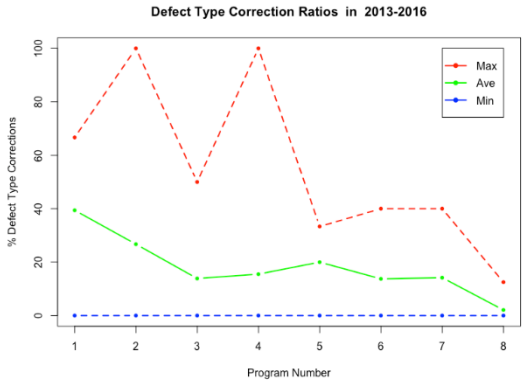


図 4　欠陥種別の修正割合の推移

4．おわりに

九州工業大学における PSP コースの受講者は、生産性を低下させることなくテスト欠陥の少ない高品質なソフトウェアを開発できている。このことから、高品質ソフトウェアの開発に必要な知識とスキルの修得に PSP が有用であることを示した。また、PSP による自己改善には、PSP インストラクタの指導が重要な役割を担っており、PSP 関連図書等を用いた自学自習では必ずしも十分な改善効果を見込めない恐れがあることも示した。

A．参考情報

[1] Humphrey, W. S.: A Self-Improvement Process for Software Engineers, Addison-Wesley (2005).（邦訳：PSP ガイドブックソフトウェアエンジニア自己改善，翔泳社, 2007 年）

[2] Humphrey, W. S.: Introduction to the Team Software Process, Addison-Wesley (1999).（邦訳：TSPi ガイドブック，翔泳社, 2008 年）

[3] 秋山他：九州工業大学におけるパーソナルソフトウェアプロセス教育-ソフトウェア品質向上のためのスキル修得-，SEC journal, Vol. 6, No. 3, pp. 118–125 (2010).

[4] 梅田政信，片峯恵一，PSP/TSP による実践的な ICT 人材育成の取り組み，情報処理学会誌, Vol. 53, No. 10, pp. 1084-1087 (2012).

2A1「課題管理からアプローチするトラブルプロジェクトの立て直し」飯田秀樹（オープンストリーム）

<タイトル>

課題管理からアプローチするトラブルプロジェクトの立て直し

<サブタイトル>

<発表者>

氏名（ふりがな）：飯田 秀樹（いいだ ひでき）
所属： 株式会社オープンストリーム システムインテグレーション事業部 戦略技術推進本部 副本部長

<共同執筆者>

氏名（ふりがな）：福良 智明（ふくら ともあき）
所属： 株式会社オープンストリーム システムインテグレーション事業部 戦略技術推進本部

<主張したい点>

トラブルプロジェクトの立て直しを行う際には、いきなり大掛かりなプロセス改善を行うのではなく、課題を整理することがプロジェクトを健全な状態に戻す為に有効であることが分かった。

<キーワード>

トラブルプロジェクトの立て直し チケット管理 課題管理 棚卸し

<想定する聴衆>

プロジェクトマネージャ
テストマネージャ
トラブルプロジェクトのヘルプに借り出されたことのある方

＜活動時期＞

2013 年 10 月～継続中

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（ ）

<発表内容>

1.背景

社内で、いわゆるトラブルプロジェクトの立て直しやプロジェクト支援を何度か行う機会があった。
その際に、まず現状 PJ 状況を把握する手段として、課題の棚卸しを行ってみた。いくつかのトラブル案件で、同じことを行う内に課題管理において共通してみられる現象があった。
課題は、個々の「チケット」としてチケット管理されている。
そこから、今回の課題管理からアプローチするトラブルプロジェクトの立て直し方法を検討し、実践してみた。

2.改善したいこと

トラブル案件で共通してみられる下記の問題

- チーム内外でのコミュニケーション不全：問題、課題発生時の報連相ができていない状態
- 成果と終わりのみえない作業：必要な作業、優先度が不明でプロジェクトメンバの士気が下がっている状態
- 制御不能に陥ったプロジェクトチーム：PM/PL の指揮命令系が崩壊し、自己判断に頼っている状態

3.改善策を導き出した経緯

トラブル案件にアサインされ、状況把握の為に課題の管理状況をみてみると下記のような状況が共通してみられた。

- 大量のゴミ課題(チケット)があふれている。ゴミ課題の特徴としては、課題の他にリスクやメモ、備忘録が分類されず混在している
- 担当者、解決方法、期限が定義されていない課題がいつまでも放置され続け、かつ次々に増えている
- 重複した課題が散在し、管理不能な状況になっている

これらは、正常なプロジェクトでは当たり前のようにできていることであり、かつ実現する為に特別なスキルが必要という訳ではない。
そこで、まずは当たり前にできていたことをできる状況をつくろうということで、この課題管理から見えてきた問題に取り組むこととなった。また、この取り組みは 1 人でも始めることができ、導入障壁が低いことも取り組んだ理由の 1 つである。

4.改善策の内容

課題管理から見えてきた問題点に対する改善策として、下記の活動を行った

- 課題の棚卸しを行う為の専任者の任命
- 課題の運用ルールの定義
- 不要な課題を排除する断・捨・離
- 上記活動の徹底

5.改善策の実現方法

- 課題の棚卸しを行う為の専任者の任命
  PM/PL は忙殺されている状況なので、外部から専任者を任命する
  専任者はあえて業務には踏み込まず、客観的な立場で、課題管理に集中する(業務知識が 0 でも問題ない)
  ただし、問題を問題と認識できるようにマネージメント業務の経験者であることが望ましい

・ 課題の運用ルールの定義
- コミュニケーション不全の状況の改善
→ コミュニケーションルールを整備し、課題の担当者や周知すべきメンバを明確にする
その際に必ず課題管理専任者が把握できるようなルールにする
- 成果と終わりのみえない作業
→ 課題起票時のルールとライフサイクルを定義する
特にどうなればその課題は解決されたと判断できるのか、終了基準を明確にする
終了基準はなるべく簡潔にし、解決しやすいようにする
そのことによって、課題があれば解決するという意識付けをメンバに対して行う
- 制御不能に陥ったプロジェクトチーム
→ 最低限記述ルールの徹底し、担当者を明確にする（5W1H で（いつ、どこで、だれが、なにを、なぜ、どうやって、～））

・ 不要な課題を排除する断・捨・離
課題ではないメモや備忘録などのゴミ(ノイズ)を排除し、整理する
「断」・・・運用ルールが決まるまでは課題の起票を制限する
「捨」・・・一定期間が経過したものは捨ててしまう。必要であれば再度運用ルールに従い起票する
「離」・・・担当者が設定されていない課題は、課題管理専任者に振りなおし、仕切り直しを行う

・ 上記活動の徹底
- 運用面
✧ 課題管理専任者は日々運用ルールが守られているか確認し、必要に応じてルールの徹底を行う
✧ メンバ担当状況を把握し、運用が滞らないように整理する
✧ 棚卸ミーティングの実施、リーダー格と課題の読み合わせを行う
✧ 担当の認識違いや優先度の違いなどを会議で調整する
✧ PM/PL とプロジェクトメンバの橋渡し的な役目も担う
- 見える化
✧ バーンダウンなどで課題件数を管理し「見える化」を行う、メンバがゴールを意識できる状態を作る

6.改善による変化や効果

2 であげた課題に対して、下記のような効果がみられた

・ チーム内外でのコミュニケーション不全
- 透明性が確保でき、誰が(自分が)何をしているのかが明確になった
- PM/PL が不在な場合でも、「今誰が何をしていて、何で課題があるか」を皆で共有できている状態を構築できた
- 課題管理を通じて、課題管理専任者から PJ における作業量の偏りや問題を検出しリーダーへの進言、メンバへのフォローができた

・ 成果と終わりのみえない作業
- ゴール設定と見える化ができた
- 誰が何をいつまでにやるかがわかるようになった
- ゴールが見えることによって、メンバのモチベーションを維持することができた

・ 制御不能に陥ったプロジェクトチーム
- 見える化に伴い、円滑な作業指示ができるようになった
- タスクの平準化による不公平感の排除できた

7.改善活動の妥当性確認

普通の状態であれば出来ていたことができなくなっている状態を解消するという点では、この課題管理の視点からの各施策は効果があった。実際に複数のプロジェクトにおいて、トラブルなく無事納品まで行うことができた。

また、課題管理専任者が１人でも施策を実施できること、高価なコンサル料も必要ないこと、ドラスティックな業務改善を伴わないことから、非常に費用対効果が高い方法だと考えている。

2A2「デザイン思考を活用したプロセス改善」服部悦子（住友電工情報システム）

<タイトル>

デザイン思考を活用したプロセス改善

<サブタイトル>

<発表者>

氏名（ふりがな）：服部　悦子　（はっとり　えつこ）
所属：　住友電工情報システム株式会社　QCD 改善推進部　品質改善推進グループ

<共同執筆者>

氏名（ふりがな）：
所属：

<主張したい点>

近年再注目されているデザイン思考はイノベーションを実現する方法論で、ユーザー視点で課題を捉えて見える化する、メンバー間で共有しながら小さな失敗から学習を素早く繰り返す、ということが特徴とされている。この手法はプロセス改善にも効果があることがわかり、具体的な活用事例と共に紹介する。

<キーワード>

デザイン思考、プロセス改善、ワーキンググループ、アイデア

<想定する聴衆>

プロセス改善推進者、チームリーダー

<活動時期>

2016 年 5 月～現在

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（　　）

<発表内容>

1.背景

当組織ではワーキンググループを中心しとした改善活動が定着しており、毎年 10 ほどのワーキンググループを立ち上げ、組織目標達成に取り組んでいる。各グループは月２～４回程度のミーティングを実施しメンバーの参加率も悪くはないが、成果が出ない、もしくは成果が出るまでに２年近くかかってしまうという課題を抱えていた。

2.改善したいこと

各ワーキンググループが１年で何らかの成果を出せるようにしたい。

3.改善策を導き出した経緯

改善活動は活発でありながらなぜ成果を出せないのか、社内の改善活動に参加しているメンバー数名に意見を聞いたところ、次の２点の課題となっているのではないかと考えた。

課題１．改善策のアイデアが出ない・出せない

品質を上げる、コストを下げるなど目標は明確だが、それをどうやって実現するのかというアイデアが出ない。

課題２．改善策がブレる

アイデアが出ても、実現していく過程で、WG メンバーの思いつきや目先の課題に囚われ
本来の解消すべき問題とズレた策になってしまう。

まず課題１については改善対象である、我々開発者自身、あるいは対象の作業への理解が不足しているのではないかと考えた。さらに自分達自身が普段業務として行っていることであり、理解が不足していることに気付かず、理解を深めるためのアクションが必要だとも思っていなかった。

次に課題２については進め方に問題があると考えた。それまでの進め方は改善策が決まれば、複数名で作業を分担できるように開発標準の作成や教育資料の作成など改善策の実現に必要なタスクを洗い出して担当者に割り振り、作成し、ワーキンググループの場でレビューするやり方をしていた。しかし、作成者やレビューに参加するメンバー間で改善対象や課題に対する理解が共有できておらず、それぞれの経験や個人の思い込みの重要度で判断してしまう。よって当初の課題とはブレた策を実現してしまう。

これらを解消するための手段としてデザイン思考に着目した。

デザイン思考とは、観察(オブザベーション)、発想(アイディエーション)、試作(プロトタイピング)を何度も繰り返しながらチームで協創するイノベーティブな活動（表 1）で、ユーザー視点で課題を捉えて見える化する、メンバー間で共有しながら小さな失敗から学習を素早く繰り返す、ということが特徴とされている。これはエンドユーザーへのソリューションだけでなく我々自身の改善活動にも使えるのではないかと考えた。

表 1 デザイン思考とは　　※参考文献[1]より

観察(オブザベーション)	発想(アイディエーション)	試作(プロトタイピング)
強い仮説にとらわれず **「無意識の声」**を聞く。 主観的に感じて **インサイト(気付き)**を得る。 **質的な活動を重視**。	**ブレインストーミング**などを活用し、 **チームが協働する**ことによって 生み出される**「集合知」を重視**。	短時間に多くのアイデアを試し改良する活動。 **頭ではなく、** **手で考える、** **体で考える。**

4.改善策の内容

デザイン思考は様々なプラクティスがあるが、今回は①ペルソナ、②AsIs シナリオ、③プロトタイプを利用した。一般的にデザイン思考では何からのソリューションを考えるにあたり、これらを順に使って検討するが、今回は改善活動の状況に応じて、効果がありそ

うなプラクティスを選択して利用した。それぞれの内容とプロセス改善活動での期待を以下に述べる。

①ペルソナ

目的に関連したユーザー像を定義し記載する。潜在的なユーザータイプを分析し、共通の属性を見つけて整理する。ペルソナを定義することにより、方向性が定まる、アイディエーションのきっかけになる、などが期待できる。[2]

＊プロセス改善活動で利用することの期待＊

改善対象となる設計者やプログラマのスキルや問題点を明確にし、ワーキンググループのメンバーで共有することにより、改善策がブレることを防げるのではないか。また施策の妥当性確認にも使えるのではないか。

②AsIs シナリオ

対象となる業務や作業の手順を明記し、それぞれの手順で行うこと(Do)、考えること(Think)、感じること(Feel)を書き出す。またそれぞれの手順で魔法の杖があればどうしたいかと問いアイデアを促す。

※以下を参考にさせてもらった。

・NC デザイン＆コンサルティング株式会社のコンサルティング　・IBM 社の IBM Design Thinking 手法

＊プロセス改善活動で利用することの期待＊

アイデアが出ないときに、改善対象の業務を書き出し、考えていることや感じていることを書き加えることによって気付いていなかった問題点や改善アイデアを引き出せるのではないか。

③プロトタイプ

初期のアイデアを物理的状態で体験できるようにし、コンセプトの適用可能性を評価する。人々に具体物を与え反応してもらうことで代替案や修正点を検討するきっかけとなり、新たなアイデアが得られることもある。[2]

＊プロセス改善活動で利用することの期待＊

システム開発で作成する外部仕様書の画面イメージほど綺麗なものでなく、時間をかけずに素早く作り、試してみる。私達はどうでしも Excel、PowerPoint などの電子媒体で無意識のうちに時間をかけ綺麗に作り込んでしまう。手書きでちゃちゃっと時間をかけずに、効果を確認できることを実現したい。

課題に対する各プラクティスを利用することの狙いについて表 2 にまとめる。

表 2 課題に対する各プラクティスの狙い

課題 / プラクティス	課題１. 改善策のアイデアが出ない・出せない	課題２. 改善策がブレる
①ペルソナ定義	方向性が定まりアイディエーションのきっかけになる	ユーザー像を定義することにより方向性が定まりブレを防止できる
②AsIs シナリオ	作業手順のそれぞれについて「考えること」「感じること」「魔法の杖」でアイデアを促す	作業手順をメンバーで共有することによりブレを防止できる
③プロトタイプ	具体物に対する反応からアイデアを得る	具体物を作って改善策を点検できる

5.改善策の実現方法

以下に 3 つの事例「5.1　UT 設計レビュー効率向上」「5.2　プログラム設計の品質向上」「5.3　ワーキンググループへ改善活動に必要な組織実績の提供」について、デザイン思考のプラクティスをどのように利用したか、どのような効果があったかを説明する。

5.1 UT 設計レビュー効率向上

(a) 背景

このワーキンググループでは UT 仕様書レビュー時間の削減を狙い、レビュー用 UT 仕様書の作成を検討していた。しかし、それが本当に時間削減に繋がるのかメンバーが自信を持てない状況であった。デザイン思考のプラクティスを使えば、事前に効果があるか評価できるのではないかと考え、メンバーに提案し実践した。

(b) 実現方法

以下に利用したプラクティス、所要時間、参加人数と具体的な実施手順を説明する。

利用したプラクティス	AsIs シナリオ、プロトタイプ
所要時間、参加人数	2 時間(８名)×3 回

・実施手順

1) UT 仕様書レビューのタスクを洗い出し、行うこと(Do)として書き出す。
2) その時に使う画面や機能も書き出す。システムの機能だけでなくメールや Excel などのツール類もあげる。
3) 参加メンバー全員でタスクの順序や利用する機能を合意する。
4) それぞれの行うことに対して考えること(Think)、感じること(Feel)の意見を出し合う。この時に気分が良くなる所、落ち込む所を色分けする工夫をした。
5) 気分が落ち込む所に着目し、「どんなことができれば解消できそうですか？魔法の杖がある前提で実現できなさそうなものでもいいですよ。」とメンバーに問いかけ、アイデアを出してもらう。
6) 気分が良くなる所はどういうことがそうさせるのか具体例を聞き、誰でもがそれを体験できるようにするにはどうすればいいかメンバーにアイデアを求める。
7) 全体的にアイデアが出つくしたら、何が実現できそうかを考える。実現不可のアイデアは、「このまま実現するのは難しいけど、これくらいだったらできるというものはないですか」と問いかける。
8) 実現できそうなアイデアの内、目標(レビュー時間削減)に寄与するかどうかを確認し、絞り込む。
9) 絞り込んだアイデアを具体化するために、プロトタイプを作成する。「時間をかけずに手書きでいいよ」と言い、2～3 名のミニグループに宿題としてお願いした。
10) プロトタイプをレビューして、実現する策を決める。

行うこと（Do）	使う画面・機能	考えること（Think）	感じること（Feel）	魔法の杖(こんなことができたら)
UT 仕様書のレビュー依頼を受ける	口頭・メール	スケジュールを考える 期日はいつか	・仕事が増えた	→ 達成感が出るものがあればいい。 → 自身の評価につながることが見えればうれしいかも（例：検出欠陥数の 社内ランキング、誤字脱字検出マスター、SQLマスター、○○マスター）
レビュー対象の UT 仕様書を開く	・ドキュメントサーバー(ブラウザ) ・Excel		・UT 仕様書を探すのが面倒だ ・レビュー対象が正しく添付されているだろうか（不安）	→ レビュー依頼のメールに UT 仕様書のリンク(所在)があると嬉しい → レビュー依頼時に自動でチェックしてくれる（添付ファイルのファイル名でチェック）
レビュー対象プログラムの外部仕様書・PG仕様書を開く ・仕様書フォルダから機能名で探して開く 又は ・プログラム文書から先行文書をたどって開く	・ドキュメントサーバー(ブラウザ) ・Excel	Revがあっているのか		

図 2 UT 仕様書レビューの AsIs シナリオ実践例（抜粋）

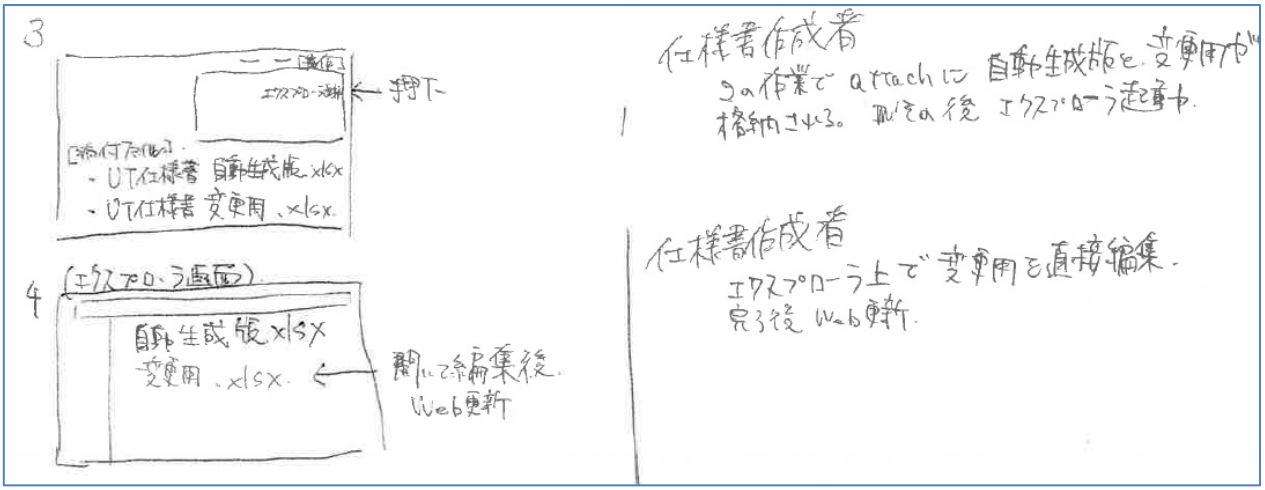


図 3 手書きによるプロトタイプの実践例

(c)結果

実践したところ、以下 4 つの効果が確認できた。

・「面倒なことはない」という思い込みの排除 <効果 1>

当組織ではどのプロジェクトも共通の文書管理システムを利用しており構成管理も明確で、単体テスト仕様書やテスト結果が保管される場所は決まっており、探す手間は無いという思い込みがあった。しかし、AsIs シナリオを使って「感じていること」を書き出すと、探すのが手間で依頼時に検証対象の所在(社内サーバーの URL）のリンクをメールで送ってもらえるとうれしい、という意見が出た。詳しく聞いてみると担当者はシステム保守業務も担当しており、そのときの仕事の状況に応じて複数システムの文書環境を閲覧することになる。また、検証対象のプログラム一覧までたどり着いても、一覧には似たようなプログラムが複数存在することもとあり、そこから対象のプログラムを探すことは面倒な感覚があった。

・「変えられない」という思い込みの排除 <効果 2>

プロトタイプを作成しワーキンググループでレビューした時に、レビュー用 UT 仕様書の作成が文書管理システムのメニューからできれば、いちいち別のシステムを立ち上げる必要はなく手間が一つ減らせることができるというアイデアが出た。UT 仕様書のレビュワーはレビューした結果を文書管理システムの所定の場所に保管する。プロトタイプを使ってウォークスルー的に評価していたときに、レビューが終わってから文書管理システムへアクセスし保管するなら最初からそこに生成すればいいのではないかというアイデアが出た。

今まであれば、「文書管理システムは社内共通の仕組みだから変えられない」という思いから、アイデアとして形にする前に対象から外してしまっていた。今回の取り組みでは、魔法の杖というキーワードで実現可否の縛りを外して考えることができ、プロトタイプは紙に書いてみるだけだからロスは無い(少ない)ので見える形にするというアクションに繋がった。見える形になれば今行っている作業の繁雑さにも気づける。見えるものができ、それをメンバーで共有し効果があると思えたら、変えることができないか調整や交渉といったアクションに繋がる。調整や交渉の材料として AsIs シナリオで洗い出した課題やプロトタイプとして効果が出ることを具体的に説明できることの価値は大きい。「変えられない」という思いより「変えることができれば効果がでる」という思いが強くなり行動に繋がる。

・アイデアが出やすい <効果 3>

いきなり「面倒なことはないですか」と聞かれても、よっぽど普段から強く思っていない限り意見は出ない(図 4)。AsIs シナリオを使って手順を共有してから、スコープを絞って「この時はどんなことを考えますか、どんな風に感じますか」と聞かれると「そういえばこんなことがあって・・・」と出やすくなる(図 5)。加えて、メンバー全員がその手順を共有して議論しているので、他の人の意見に共感できたり、わからないことは質問したりと課題の明確化に繋がる。

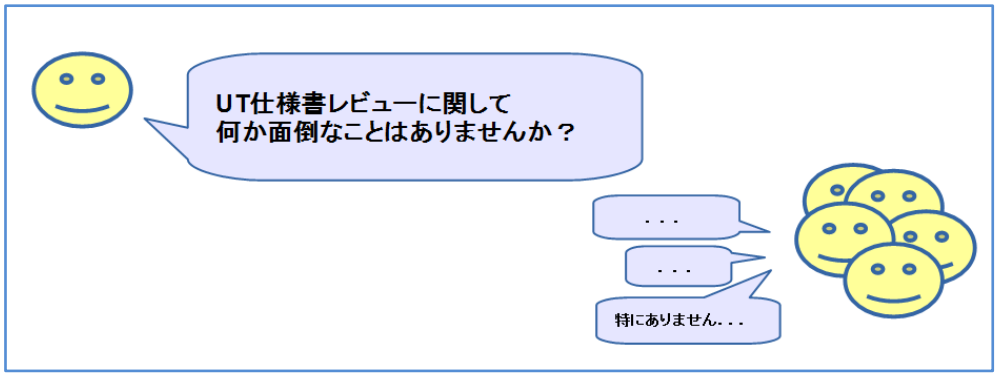


図 4 今までのアイデアの引き出し

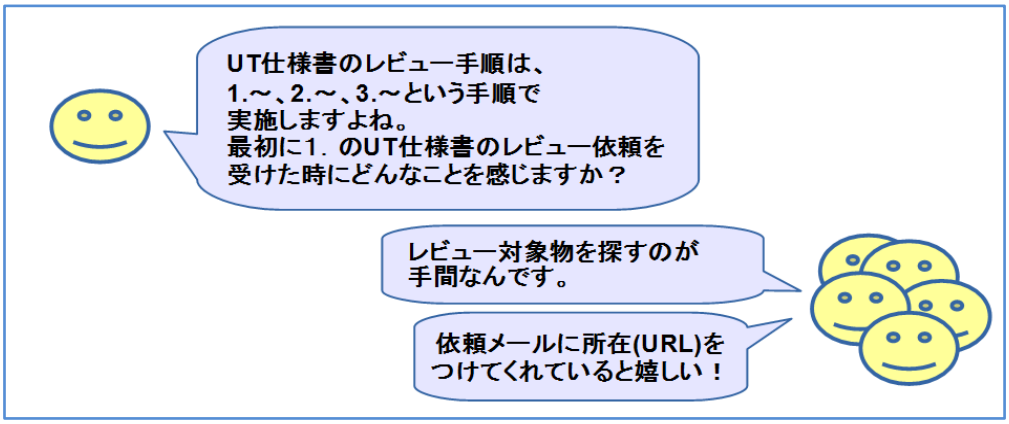


図 5 デザイン思考(AsIs シナリオ)を使った時のアイデアの引き出し

・メンバーの改善意欲向上 ＜効果 4＞

このワーキンググループでは、毎回、会議の最後に KPT 形式で振り返りを行っている。この取り組みを実践したときの振り返りでは Keep(よかったこと、続けること)として、「魔法の杖の検討で面白い案が多く出た」「差分版 UT 仕様書のイメージがわいてよかった」「新しい取り組みを知ることができた」などの意見があがり、改善活動に対する意欲向上に繋がった。

5.2 プログラム設計の品質向上

(a)背景

プログラム設計の品質向上に向け、プログラム設計プロセスを見直そうとしていたが、ワーキンググループのメンバーから出る意見は、「プログラマが初心者の場合、仕様書としてここまで書いておかないとプログラムの品質が担保できない」というものであった。プログラム設計の品質を良くする施策を考えたいのに、初心者プログラマのことが気になりプログラム設計の検討が進まない状況であった。プログラム設計の議論へ集中できるように想定するプログラマのレベルを明確にすればいいのではないかと考え、ペルソナを定義してみることを提案し実践した。

(b)実現方法

以下に利用したプラクティス、所要時間、参加人数と具体的な実施手順を説明する。

利用したプラクティス	ペルソナ
所要時間、参加人数	2 時間(5 名)×2 回

・実施手順

1) PG 仕様書を利用するプログラマのスキル（データモデリング、RDBMS、Java、・・・etc.）を洗い出す。
2) プログラマがそれぞれの属性でどの程度のスキルを持っているかを決める。（○○ができる、○○が判断できる）今回は初心者向けの施策を考えたいわけではないので、中級者以上をイメージして決めた。（図 5）

想定するプログラマのレベル

■ データモデリング
・E R 図が読める
－テーブル間の関係が読める（1 対多、絶対存在する/存在しないことがある(白丸)…）
－標準的なデータの流れを読める（左から右へ、上から下へデータができる）
－プライマリーキーと論理キー(受注 ID と受注 NO の違い)を区別できる
－従事するプロジェクトの ER 図の全体（担当サブシステム）をある程度把握している
：
■ RDBMS
・SEI/SIS 独自ルール
Exist Key, LatestKey (Null Key)
－NK、LK を理解している。
：

図 6 中級者以上を想定したプログラマのペルソナ（抜粋）

(c)結果

実践したところ、「施策のブレを防止できる」という効果を確認できた。

・施策のブレを防止 ＜効果 5＞

図 5 のようにペルソナ定義を作成しメンバーで共有した。これで議論は PG 設計に集中できると思ったが、それでもメンバーの思いはどうしても目の前にいる若手や経験の浅いメンバーに対する課題が解消できなかった。メンバーにどこが課題なのか聞いたところ、”従事するプロジェクトの ER 図の全体（担当サブシステム）をある程度把握している”ところだということがわかった。具体的には、仕様を実現するデータをどのように抽出するか、対象テーブルの特定やその結合条件について ER 図から読み取れない人がおり、細かく指示しないと結局プログラムに手戻りが発生することになる。そこの検討は不要だと言われても不安が残る、という強い思いがあることがわかった。

この問題を解消しないと議論が進まないことがわかり、先に初心者プログラマの対応を決めることにした。メンバーと相談し、プログラム設計プロセスを 2 種類用意することにした。具体的には開発計画を立案するときに、予定するプログラマのスキルにあわせてどちらのプログラム設計プロセスを適用するのか決定することにした。これを業務フローとして図示し共有することでメンバーは安心して、プログラム設計プロセスでやるべきことの検討に集中できた。

ペルソナを定義したことで、初心者プログラマ向けの施策にブレてしまうことを防止できた。また、メンバーがどこに不安を持っているかが明確になり、その対策案を先に決めることで検討すべき課題に集中できた。

5.3 ワーキンググループへ改善活動に必要な組織実績の提供

(a)背景

私の所属する品質改善推進グループでは、ワーキンググループが改善成果を出せるための支援策を検討していた。改善活動に必要な仮説の検証や評価のために定量的データが必要となるが、組織のデータとして必ずしも利用しやすい状態でまとまっているわけではなく、また改善活動に参加している現場の開発者はどのようなものが存在するかも知らない人が多かった。よって組織としてどういうデータが存在するか、一目でわかる HP があればいいのではないかと考えた。

HP の見せ方を具体化しようとしたときに、改善推進者 2 名はアイデアがわかず何からどう決めればよいかわからない状態であった。そこで AsIs シナリオを利用すれば、各ワーキンググループがデータを必要とするシチュエーションを具体的にイメージでき、どのような提供手段をとればいいかアイデアが出せるのではと考えた。

(b)実現方法

以下に利用したプラクティス、所要時間、参加人数と具体的な実施手順を説明する。

利用したプラクティス	AsIs シナリオ
所要時間、参加人数	2時間(3名)×1回

・実施手順

1) ワーキンググループが行う改善活動の手順を洗い出す。（目標設定、現状調査・問題分析、・・・）
2) 定量データを使うシチュエーションを具体的に書き出す。

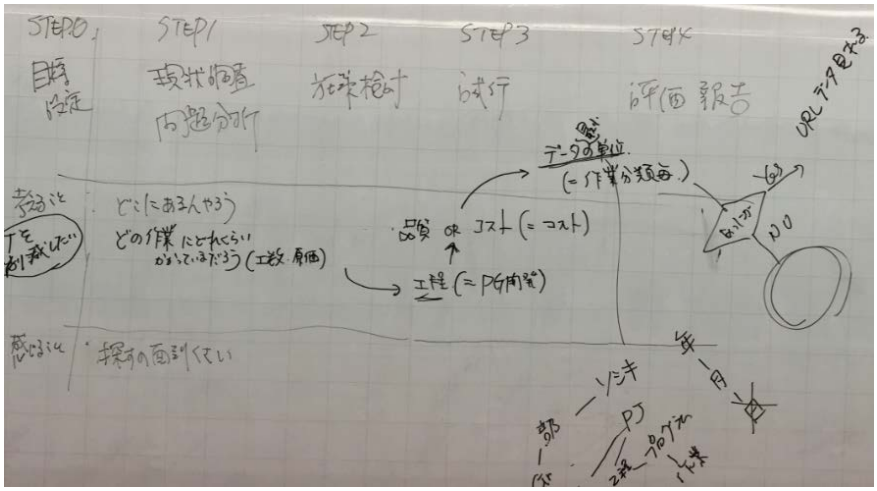


図 7 AsIs シナリオの例

(c)結果

実践したところ、「アイデアが出せる（0 から 1）」という効果を確認できた。

・アイデアが出せる(0 から 1) ＜効果 6＞

プログラム開発の生産性向上を検討するワーキンググループを例にあげ、活動を開始したときにどのようなことを考えるかイメージを書きだしてもらった。そうすると、「最初は、どの作業にどれくらいかかっているだろう、と思うね。」、「だとすると、データの一覧があれば、工程＝PG 開発、品質/コスト＝コスト と絞り込めればいいね。」「さらにデータの期間や粒度などを知りたいね」というように自然にイメージが湧いてきた。これを何種類かのワーキンググループやシチュエーションを想定して繰り返し、どのように見せればよいか具体的に決めることができた。

AsIs シナリオを利用することによって、当初何をどう決めればよいかわからないと言っていたメンバーが、見せ方のアイデアを具体化することができた。

6.改善による変化や効果

当初の課題に対してデザイン思考の各プラクティスで狙ったことに対する評価を表 3 にまとめる。またプロセス改善にデザイン思考を利用することの価値を表 4 にまとめる。

表 3 課題に対する各プラクティスで狙ったことの評価

課題 / プラクティス	課題１. 改善策のアイデアが出ない・出せない	課題２. 改善策がブレる
①ペルソナ定義	方向性が定まりアイディエーションのきっかけになる ⇒◎ **（効果５）**	ユーザー像を定義することにより方向性が定まりブレを防止できる ⇒◎ **（効果５）**
②AsIs シナリオ	作業手順のそれぞれについて考えること、感じること、魔法の杖でアイデアを促す ⇒◎ **（効果１、効果３、効果６）**	作業手順をメンバーで共有することによりブレを防止できる
③プロトタイプ	具体物に対する反応からアイデアを得る	具体物を作って改善策を点検できる ⇒◎ **（効果 2）**

表 4 プロセス改善にデザイン思考を利用することの価値

観察(オブザベーション)	発想(アイディエーション)	試作(プロトタイピング)
強い仮説にとらわれず **「無意識の声」**を聞く。 主観的に感じて **インサイト(気付き)**を得る。 **質的な活動を重視**。	**ブレインストーミング**などを活用し、 **チームが協働する**ことによって 生み出される**「集合知」を重視**。	短時間に多くのアイデアを試し改良する活動。 **頭ではなく、** **手で考える、** **体で考える。**
↓ プロセス改善に利用することの価値 ↓		
＜効果 1＞「面倒なことはない」という思い込みの排除 ＜効果 5＞施策のブレを防止	＜効果 3＞アイデアが出やすい ＜効果 4＞メンバーの改善意欲向上 ＜効果 6＞アイデアが出せる(0 から 1)	＜効果 2＞「変えられない」という思い込みの排除

7.改善活動の妥当性確認

当初認識した「アイデアが出ない・出せない」という課題については、AsIs シナリオやペルソナ定義のプラクティスを使うことによるアイディエーションの促進で解消できた。もう一つの「改善策がブレる」という課題については、同じプラクティスでもインサイト（気付き）を得る効果から軽減できそうである。またプロトタイプを使って改善策を検証することの効果も確認できた。どの事例も 1 回あたり 2 時間程度で実践できておりコストの問題は無い。よって今回の取り組みは妥当であると判断した。

この取り組みは始めたばかりであり、ワーキンググループが活動成果を出せるところまでは確認できていない。しかし、有用なことが確認できたので、今後、組織のプロセス改善活動の手段として展開できるよう以下対策を取り組みたい。

対策１. 改善活動を進めるメンバーが気軽に使ってみよう思えるような工夫

対策２. この手法をうまくファシリテートできる人の育成

A. 参考情報

[1] 前野隆司著, ”システム×デザイン思考で世界を変える”, 日経 BP 社, 2014

[2] ヴィジェイ・クーマー著,” 101 デザインメソッド 革新的な製品・サービスを生む「アイデアの道具箱」”,英治出版,2015

2A3「SEPG 活動の立ち上げと、組織定着への取り組み」清水崇司（ニコンイメージングシステムズ）

<タイトル>

SEPG 活動の立ち上げと、組織定着への取り組み

<サブタイトル>

10 年に亘り、「作業者にとって優しいか」を理念に活動した報告

<発表者>

氏名（ふりがな）：清水崇司（しみずたかし）
所属： 株式会社ニコンイメージングシステムズ 第一開発部 第五開発課

<共同執筆者>

氏名（ふりがな）：
所属：

<主張したい点>

・10 年にわたる改善活動（SEPG 活動）の中で経験した、イマイチだった活動を改善できた事例とともに、改善活動を継続できた要因を、組織、教育、活動、マインド、の観点で報告する。
・事例紹介では主にふりかえり活動の方法、管理シートの紹介を行う。
・他課に比べ、高い品質の成果物につながっていること。
・2017 年に全社組織として全社 SEPG が作られることになった。

<キーワード>

改善組織（SEPG）立ち上げ、ふりかえり、ルール、やる気、

<想定する聴衆>

これから改善活動に取り組む方、改善活動に取り組んでいる方

＜活動時期＞

2007 年～2017 年 3 月

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（ ）

<発表内容>

1.背景

2006 年当時、ソフトウェア開発のテスト工程において欠陥が大量に見つかり、手戻りが発生することに困っていた。

カメラの製造現場では、小集団活動など改善活動が当たり前に行われている。しかし、ソフトウェアの開発現場では改善活動と言うものが無かった。

そんな中、SPI（ソフトウェアプロセス改善）と言う世界があることを知り、JASPIC が主催する SPI JAPAN2006 に参加した。会社の枠を超えて事例が紹介され、忌憚なく意見を交わしている姿に感銘を受けたと同時に、このままでは弊社が取り残されてしまうという危機感に駆られた。自分の組織でも仕事の仕方について改善するチームを作りたいと強く思った。

そして、2007 年に職場に改善活動を行うチームを作った。

10 年にわたり改善活動（SEPG 活動）を継続できた要因を組織、教育、活動、マインド、の観点で挙げる。

【組織】

・・職場のマネージャの理解を得た

・・職場のリーダーと共に立ち上げた

・・メンバを入れ替えた

【教育】

・・初期段階で SEPG メンバ全員が、SEPG 活動を計画的に実施するという教育を受けた

・・社外の改善活動を知ることで、自社の活動に対し奮起することにつながった

・・社外の改善活動事例や、改善手法を課内に紹介し続けた

【活動：開発メンバ、マネージャを対象とした活動】

・・ふりかえり活動を推進し、プロジェクト管理業務の一部分を担う

（ふりかえり活動がプロジェクト完了報告の一部分として利用された。）

・・業務手順、規約（ルール）、成果物フォーマット、管理ツールの担い手になった

・・業務ツールのパイロット調査を行う担い手になった

【活動：SEPG 活動を活性化する仕組み】

・・社内表彰制度を利用し活動の成果をまとめるきっかけにした

・・定期的な活動

・・外部活動への参加

・・開発プロセスのルール・規約などは SEPG が管理した

・・SEPG の活動内容と社内のマネジメントシステムの対応表を作成し、SEPG 活動が社内のどこに影響を与えているかを示した

【マインド】

・・「手順」と「ツール」を、「作業者にとって優しいか」と言う観点を持って構築し改善を行った。やりやすいかということ。

・・みんなに有り難いと思われる活動を行う。（プロセスモデルの押しつけでなく。）

・・何でもやる姿勢。業務に関わることなら何でもいったん引き取る。

・・プロセス改善について幅広い知見を持ち、理由と自信を持って改善活動を行う。

この 10 年間で 60 件程の取り組みを行ってきた。
以下に SEPG 活動で取り組んだ事例をいくつか紹介する。

2.改善したいこと
（1）結合テストフェーズで欠陥が頻発する。
（2）各チームの各開発フェーズの成果物の質がわからない。
（3）課題管理システムが変更される予定であり、運用が混乱するリスクがある
（4）プロジェクト完了時にふりかえる機会が無い。
（5）SEPG 活動の価値を周囲に示せていない

3.改善策を導き出した経緯
（1'）設計やコーディングフェーズで欠陥が見つける仕組みが弱い。
（2'）フェーズ区分定義、設計、コーディングフェーズ、テスト工程の欠陥数をカウントする方法が無かった。
単体テストツールを使用していたが、費用対効果が悪く、非効率な作業になっていた。
（3'）業務サポートツールの担い手になることで、不便だったり困ったりすることをいち早く察知することができる。
（4'）ふりかえりを行う際には、ふりかえるネタと、起票するフォーマットを作成する必要がある。
（5'）SEPG 活動は QCD に影響を与えている。QCD 以外に組織の活動に影響を与えていると言える示し方はできないだろうか？と考えた。

5.改善策の実現方法
（1''）レビュー方法の構築。2 段階レビューの導入。（構成や文章表現の確認と、内容の確認を分けた。）
（2''）詳細なフェーズ区分定義を行い工数入力の精度を向上し、議事録システムや試験帳票から欠陥数を集計する方法を構築した。
規模測定の自動化ツールの設定について定義した。
単体テストを効率よく行うチェックシートのフォーマットを作成し、導入した。
（3''）課題管理チケットシステム導入の際にパイロット運用を行った。また、不具合分析の選択項目について定義した。
（4''）ふりかえるネタ(数種の品質分析グラフ)と、改善施策を記入しプロジェクト横断で管理するフォーマットを作成した。そして、ふりかえる機会を設け、推進活動を行った。
（5''）SEPG の活動内容と社内のマネジメントシステムの各項目との対応表を作成し、SEPG 活動が社内の活動のどこに影響を与えているかを示した。

6.改善による変化や効果
（1'''）構成や文章表現の確認で大きく手戻が発生することが無くなった。
単体テストを効率よく行えるようになった。
（2'''）欠陥数や規模を集計する手間を削減できた。
（3'''）課題管理チケットシステム導入による混乱は無く運用された。他課では運用されない部署もあった。
（4'''）ふりかえりを実施したが、改善施策は挙げたまま実施されることは無かった。ここで、ふりかえり活動は「推進しないと消えてしまう」と言うことがわかった。ふりかえり活動をデザインし直した。新しいふりかえり活動は改善施策を次のどのプロジェクトで導入するかを宣言してもらうものであり、宣言したプロジェクトのチケットに起票し、改善施策の実施を、SEPG が促すものである。この活動によってやっと、ふりかえりでの改善施策を行うことが習慣化され、プロジェクト完了報告を作成する際のプロセスの一部になった。
（5'''）社内のマネジメントシステムと SEPG の活動の対応表により SEPG 活動の意義を示すことに役立った。

7.改善活動の妥当性確認

・ここ数年、プロジェクトの流出欠陥密度が徐々に低くなって、目標としていた基準を下回ることに成功していること。

・10 年間、課内において信頼されるチームになったこと。

2B1「WHY に重点を置いた人材育成」片山泰司（オムロン ソーシアルソリューションズ）

<タイトル>

ＷＨＹに重点を置いた人材育成

<サブタイトル>

やりがいのあるプロセスへ

<発表者>

氏名（ふりがな）：片山 泰司（かたやま やすじ）
所属： オムロン ソーシアルソリューションズ株式会社
ソリューション事業統轄本部 公共ソリューション事業本部 システム開発部

<共同執筆者>

氏名（ふりがな）：
所属：

<主張したい点>

品質をより良くするには、品質を良くしたいと考えて、実際に行動まですする人材を増やす必要がある。プロセスとは何か、何のために必要なのか、を現場に納得してもらい、プロセス定着による品質確保につなげることを狙って活動に取り組んだ。ＷＨＹに重点を置いた、新たなトレーニングを実施し、プロセスの目的や意義を伝えることで、プロセスに対する現場の納得感や関心を高めた。取り組みとして、取り入れやすい内容と考える。

<キーワード>

人材育成、トレーニング、モチベーション

<想定する聴衆>

ＳＥＰＧ、設計・開発部門の方、品質管理・品質保証部門の方

＜活動時期＞

２０１６年５月から、現在も継続中

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
■ 改善活動を実施したが、結果はまだ明確ではない段階
□ 改善活動の結果が明確になっている段階
□ その他（ ）

<発表内容>

1.背景

本発表の対象となる活動に取り組んだ主体は、弊社の鉄道ソリューション事業のシステム開発部門である。本取組みの経緯とそのきっかけを、次に述べる。組織の取り組みの一つに「当たり前のことを当たり前に実行」があった。プロジェクトを進めるために必要な「当たり前のこと」＝「プロセス」を理解し、品質向上のために、積極的にプロセスを活用してもらえるよう、新たなプロセストレーニングに取り組んだ。トレーニングの対象プロセスは、エンジニアリング領域。

2.改善したいこと

改善したいことは「プロセスに対する納得不足」→「プロセス不遵守の増加」→「市場不具合の増加」のサイクル。この負のサイクルの歯止めとして、まずはプロセスに対する現場の納得感を高めることに取り組んだ。

3.改善策を導き出した経緯

品質をよりよくするために、品質を良くしたいと考えて、実際に行動まで出来る人財を増やす必要がある。その考えのもと、人財育成に取り組んだ。過去のプロセス理解度アンケートから、現場の納得感を高めれば、プロセスが腹落ちし、実態として現場が出来ている、モチベーションが高い状態につながると考えた。現場の納得感を高めるため、トレーニング実行、プロセスの品揃え拡張、改善効果の見える化、プロセスの狙い・目的の理解促進、を取り組み候補にあげた。その中から、プロセスの目的（ＷＨＹ）の説明を中心とする、新たなプロセストレーニングを検討した。

4.改善策の内容

過去のプロセストレーニングが、ＷＨＡＴやＨＯＷの説明が中心だったり、また一度は全体的なトレーニングを実施したものの、その後はプロセス改定部分の差分トレーニングの実施が中心だったりで、新人や異動者に対するトレーニングが不足していた。今回、新人からベテランまで、プロセスの目的・必要性に納得してもらえるよう、新たなトレーニングの工夫に取り組んだ。過去のトレーニングでは手薄だった、プロセスとは何か、何のために必要なのか、の目的や意義の話を充実させたり、プロセスの何が嬉しいのかの話、プロセスに関する社内外の取り組みの話、やりがいを感じてもらうための話、の説明を新たに取り入れた。

5.改善策の実現方法

トレーニング内容を、以下の構成で新たに作り直した。

（１） プロセスに関する一般的な話
（２） プロセスの効用の話
（３） 自社や他社の取り組みの話
（４） 自社の標準プロセスの話
（５） 心に火をつける話

過去のトレーニングは（４）の説明で、内容もＨＯＷが中心だった。今回（１）（２）（３）（５）を新たに取り入れ、（４）もＷＨＹの説明が中心となるように、教材を作り直した。（１）（２）は、プロセス改善ナビゲーションガイド ～なぜなに編～や、ソフトウェア品質知識体系ガイドなどの参考文献から、ＷＨＹが伝わるような内容を説明。（３）は、自社の品質状況をデータで語ったり、他社の取り組み事例も紹介することで、現場の危機感の醸成を狙った。自社の品質状況は、開催月ごとに最新実績を反映し、教材の鮮度を保った。（４）は、プロセスの目的と、どんな問題の予防になるかをセットで説明した。またベテランは知ってることが多いので、新鮮味を出すために、標準プロセスではないが、参考情報として現場で活用されているノウハウや、全社共通のノウハウも紹介した。どんな問題の予防になるのかは、不具合報告書の発生原因や本質的要因の選択肢を活用して説明した。（５）は、いくら目的や必要性を伝えても、聞いた側が心からそれをやりたいと思わないと長続きしないので、やらされ感ではなく、やりがいを感じてもらうために実施した。

6.改善による変化や効果

トレーニングの有効性はアンケートで評価した。アンケート評価項目は以下５項目（最小値１、最大値４）。

（１）目的の達成度
（２）全体的な理解度
（３）業務への適用度
（４）講師に対する満足度
（５）教材の充実度

評価は過去トレーニング結果と比べ、全項目で上昇した（各項目、平均約 0.4 ポイント UP）。ＷＨＹの説明に重点を置いた成果と考える。そして任意記入の感想・要望欄も、プロセスに対する納得感や関心度合いとして着目した。記入率は８０％で、高いのではないかと考える（過去トレーニングは３５％）。感想・要望のうち理解・納得や品質意識が向上したと思える前向きなコメント数から、約半数の人がプロセスに関心を持ち、かつ前向きにとらえてくれたと考える。またアンケートのテキストマイニング結果（共起ネットワーク図）からも、「プロセス」・「理解」、「必要性」・「理解」の出現があることから、プロセスへの納得や関心に効果があったと考える。

7.改善活動の妥当性確認

アンケート評価項目の結果と、任意記入の記入率・内容から、現場の納得感や関心は増したのではないかと考える。これがプロセスの定着につながると考えている。アンケート回答により、今後のトレーニングニーズも収集することができた。今回はエンジニアリング領域でのプロセストレーニングであった。今後はプロジェクト管理領域など、他の領域のトレーニングを改善したり、他部門への展開も進めていく。納得感や関心が、実際に行動にまでつながったのか、プロセスの定着度合いも測定していく。

A. 参考情報

[1] ソフトウェア品質知識体系ガイド －SQuBOK Guide－ SQuBOK 策定部会 2007
[2] 芝本秀徳の『プロジェクトマネジメントの守破離』 プロセスって何それ？食べれるの？ 2012
http://blogs.itmedia.co.jp/hideshibamoto/2012/07/post-8b1d.html
[3] 壊れ窓理論の経済学 マイケル・レヴィン 2006
[4] トヨタの強さはクレドにある - INSIGHT NOW!プロフェッショナル 野町 直弘 2008
https://www.insightnow.jp/article/2132
[5]「仕事の意義」を実感させて、部下のやる気を引き出す秘訣 目標の魅力を高める「ラダー効果」「サンクス効果」
小笹 芳央 2009 http://www.nikkeibp.co.jp/article/nba/20090105/181861/
[6] PHP 文庫『子どもが育つ魔法の言葉』 ドロシー・ロー・ノルト著 レイチャル・ハリス著 石井千春訳 1999
[7] OMRON BASICS オムロン株式会社 グローバル人財総務本部 2017

2B2「人に満足を与えるものづくり・コトづくりのマインド改革」島林大祐 (富士通)

<タイトル>

人に満足を与えるものづくり・コトづくりへのマインド改革

<サブタイトル>

品質特性ワークショップ（アジャイル編）

<発表者>

氏名（ふりがな）：島林　大祐　（しまばやし　だいすけ）
所属：　富士通株式会社　共通ソフトウエア技術本部　自律改善推進室

<共同執筆者>

氏名（ふりがな）：
所属：

<主張したい点>

人に満足を与えるものづくりをするうえで、アジャイル開発でもウォーターフォールでもやるべきことはや、本質は同じだとおもいます。なぜ、自分たちの作っているものが売れないのか売れるためにはどうするのかを体験してもらいます。
その上でアジャイル開発がいかに顧客満足や市場満足を考えるうえでフィットするのか、お客様だけでなく、周りの人たちや上司に納得性のあるデータ（ISO 品質特性）をみせることで、アジャイル開発への理解を得ながら決める開発を進めるための体験型ワークショップを行っています。

<キーワード>

顧客満足
市場満足
品質
ISO 品質特性
決める開発
フィードバックの重要性
開発プロセスが形骸化

<想定する聴衆>

ソフトウエアエンジニア
ソフトウエア開発の幹部
アジャイル開発で上の理解が得られない開発者
プロセス改善担当者/SEPG
品質担当者/SQAG

＜活動時期＞

ウォーターフォール編　2010 年～2012 年
アジャイル編　2016 年 10 月～

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（　　）

<発表内容>

1.背景

ものづくりの現場において、市場調査、顧客フィードバックなどをもとに製品開発が行われているが、汎用性の高いミドルウエア開発においてすべての顧客ニーズを満たすものづくりは困難を極める。

現場では自分たちが得た情報を元に新製品の企画が行われるが、多くの職場では以下の悩みが後を絶たない。

・品質確保の技術やノウハウがうまく伝承されていない
・品質確保がなされて、ものが出荷されるまでが時間がかかる
・なかなか決まらない（コンセプト、機能など）
・魅力的な機能なのか？と聞かれてもうまく答えられない。
・プロダクトアウトに企画が進みがち
・製品品質で品質を語る風潮
・開発プロセスが形骸化（開発→品質→生産）

2.改善したいこと

お客様や市場の満足のいくものづくりとは何かにフォーカスすることで、魅力あるものづくりに。ISO 品質特性の利用時の品質目標を定めた上ではじめて、製品の品質目標を定めるスタイルに。お客様、開発者、品質担当者、幹部、サポート、SE、営業、などすべてのステークホルダが同じものを見てものづくりを進めることで、ほしい、うりたい、世の中のためになるものを作る（三方よし：近江商人）

KANO モデルを利用し、これから実践での活用で作ろうとしている機能が利用者にとって「魅力的」なのかどうかを事前に確認。開発元はお客様の「無関心領域」に入る機能は作らない。判断は工程の切れ間やスプリントレビュー時にステークホルダみんなでやる。

お客様からのフィードバックの重要性を体験、小さくつくってリリースすることでフィードバックはこまめにもらい、手戻りの影響度をより小さくし、要求の変化に追随し、作り直しの影響を少なくする。

3.改善策を導き出した経緯

発表者が開発をやっていたころの経験がもと。新規のビックプロジェクトだったため、社内の各方面から開発チームが集められた寄せ集めの集団だったため、我こそがソフトウエア開発の名人であるというベテランたちから右も左もまだわからない新入社員まで、ノウハウも知識量もバラバラ。名人たちは各々自分のやり方が一番正しいと思っており、名人芸を押し付けあう状態。決めないといけないものが決まらず、お互いの資料を理解するにも時間がかかっていた。

そのときに自ら SEPG として全体を束ねプロセス改善、品質確保、マインドアップを行った win-win 試作のひとつを最新化し開催してほしいと現場からオファ。現在アジャイル開発を推進している部隊にも使えるので、ある本部の幹部社員 200 名に対し実施。

種々の改善は、全員が共通の管理指標、管理簿を用いる事がキモ。単に見える化といっても各々がバラバラの見える化を行うと第三者が理解するのに時間がかかり、負のコストが増加する。互いを理解し満足を得るすなわち WinWin になるためには足並みをそろえるところから。

4.改善策の内容

ISO 品質特性を用いた WS を開催。Agile もウォーターフォールもお客様が満足するものを提供することへのマインドと、ステークホルダが互いに理解しあえるツールが必要。

WS は 1 日開催。まず、午前中は基本的な考え方や過去の失敗事例を元にミニワークを続け、KANO モデルを用い、自分たちの開発中の製品がお客様や市場にとって魅力的であるかを判定。簡単に売れないことに気づいてもらう。いかにして外にでてフィードバックを得るか、出るためには何をすべきなのか難しいことではないことを解ってもらう。

午後からは、現在あまり売れなくなってしまった工業製品を元にしてどうすれば売れるものになるのか、新製品を考えてもらう。その際 ISO 品質特性を用いて実際に業務になぞらえて体験してもらう。

5.改善策の実現方法

現場の幹部 200 名に対し、上記改善策を実施。午前中は座学とミニワーク、午後はワークショップとライトニングトーク形式で、チームビルディングや失敗事例などをも織り交ぜながら退屈しないように。

6.改善による変化や効果

今後の開発においてのチェックとして利用時品質をはじめ、魅力的であるかという判断基準がプロセスに入るようになった。サポートチームも開発と話がなかなか通じ合わなかったので品質特性を用いて会話するようにルールを変更。本ワークショップが開発元の必須教育に。今回依頼を受けた本部より 600 名の教育依頼を受けている。他本部も今後展開

7.改善活動の妥当性確認

活動後参加者全員にアンケートを実施。参加者それぞれの職場で起こっている課題は何か、それは解決できたのか？などを調査。

A. 参考情報

[1]なし

2B3「ヘルスケア商品における「自律」を主眼に置いたプロセス改善」伊達渡（オムロン ヘルスケア)

<タイトル>

ヘルスケア商品における「自律」を主眼に置いたプロセス改善

<サブタイトル>

～STE/SEPG/SQA 三位一体のプロセス改善～

<発表者>

氏名（ふりがな）：伊達 渡

所属： オムロン ヘルスケア株式会社 商品開発統轄部

<共同執筆者>

氏名（ふりがな）：

所属：

<主張したい点>

不具合一つに対し分析を行い再発防止のために手順の改定を進めるが、不具合一つ一つに再発防止プロセスを実施すると、作業過剰と工数増により新しい不具合を生む負のサイクルが生まれていた。

負のサイクルを断ち切るため、SEPG や SQA のほかに、STE(教育専門チーム)を立ち上げ、「自律」をテーマに、「開発者の力量整備と育成」、「開発手順の整備」を行い、不具合低減を目指す。

<キーワード>

人財育成、教育、力量、SEPG、SQA、自律

<想定する聴衆>

現場の教育に悩んでいるリーダー、SEPG、SQA、ソフト開発者

＜活動時期＞

2014 年４月～現在

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（ ）

<発表内容>

1.背景

ヘルスケア商品では、ソフトウェアの規模が小さいものの、種類が多く、様々な商品を提供している。そのなかで、規制当局の監査対応や IoT などつながる機器が増えてきたことによりソフトウェアの開発工数が増大し、不具合が増えていった。さらに、不具合が増えたことによる再発防止策としての手順を追加することにより、開発者の工数が増えさらに不具合が増える悪循環が蔓延していた。そのため、開発者の自律を目的とした教育を中心に行う専門チーム、STE を立ち上げ、STE/SEPG/SQA が協力して開発プロセスの改善を実施し、不具合の減少に取り組む検討を行った。

2.改善したいこと

ヘルスケア商品、開発完了後における不具合の減少に向けて以下の 2 点を改善する。

A：再発防止による過度な手順をみなおし、本質的な手順内容に修正する。

B：開発者の力量を定義し自律した人財育成をおこなう。

3.改善策を導き出した経緯

開発者の力量にはばらつきがあり、一つの手順を実施するのに、力量の高い人は説明不要で行うことができるが、力量の低い人には説明が必要となる。また、不具合を生み出すのは、力量の低い人が人に聞くことができずに仕事を進めてきた結果が多く、その再発防止策は、力量の高い人にとっては、作業工数として無駄になっていた。また、力量の高い人には、「こうすればもっとうまく仕事が進むのに」というアイデアを持っているが、現在のプロセスではそのアイデアが生かしきれていなかった。

4.改善策の内容

（やってきたこと）

「自律」した人財育成のプロセスを確立する。

A:開発者の中から信頼されている開発者を選んでもらい、教育専門者として配置を転換（STE）し教育を実施

B:力量レベルを 6 段階定義し、各個人のレベル判定を行い、開発者に自己認識をさせる

C:SEPG/SQA/STE が開発プロセスを見直す。具体的には力量の高い人の行動を軸に抽象度の高い手順を作成し、開発者自身で自分なりの開発方法を考えるような開発手順の内容に作り替える。

5.改善策の実現方法

1 ：教育を実施できるように、開発者のなかから信頼されている開発者を選び、教育専門者（STE）として配属

2 ：STE 主導のもと、開発者の力量レベル 5 段階を定義し、各開発者の力量レベルを設定する。

5 級：とりあえず開発できる　3 級：一人で開発できる　2 級：教えることができる。

3 ：STE/SEPG/SQA で、新しい手順を新規に作成

力量の高い人の行動に注目し、品質向上ために、どのような行動を行っているかを観察し、行動を手順として制定

・ 新しい手順書は、開発者への教科書となる（新人は勉強のきっかけ、ベテランには新たな気づきを与える。）

・ 手順の内容は、定量的に実施する項目を列挙することではなく、定性的に項目を列挙する。そのようにすることで、やることは明確になるが、やり方は開発者が主体的に出すことができ、開発者のモチベーションが高い状態で開発を進めることができる。

・ 具体的行動にさせるためには、STE（教育者）からフォローを受け、開発行為に生かす。

・ どのように具体的行動を行うか、どのような成果物が出るかの計画を開発者自ら下記、その計画内容をレビューし、計画の実施状況を SQA がチェックする。

・ レビューに、SQA が参加し、レビューで出たよかった点や悪かった点を都度 SEPG にフィードバックする。

6.改善による変化や効果

・過剰にあった再発防止による手順がなくなり、わかりやすくなった。

・開発者が、力量を上げるためには何をすればいいのかがわかりやすくなった。

・手順そのものを教科書ととらえて、新人などの導入教育や中途で開発に参加した人の教育につながった。

・開発者にとって気軽に相談できる相手ができた。

・自律人財育成のプロセスが確立し、スキルを持った自律した人財が増えていった。

7.改善活動の妥当性確認

最終ゴールとして開発完了後の不具合が、

・FY13 を 100%とし、FY14 で 60%　FY15,FY16 で 25%にまで減少した。

2C1「IoT に欠かせない BLE 通信のテスト自動化によるテストプロセス改善」伊藤卓也（オムロン ヘルスケア)

<タイトル>

IoT に欠かせない BLE 通信のテスト自動化によるテストプロセス改善

<サブタイトル>

～早く、確実に、誰でもできる～

<発表者>

氏名（ふりがな）：伊藤 卓也（いとう たくや）
所属： オムロン ヘルスケア(株) データヘルスケア事業部

<共同執筆者>

無し

<主張したい点>

IoT の本格的な普及にともない、弊社では、BLE(Bluetooth Low Energy)に対応した商品開発が増えてきている。しかしながら、多種多様な商品へ対応することによって、通信仕様が複雑化し、弊社の BLE 通信モジュール開発のテストが肥大化していた。その結果、BLE 通信モジュール開発が商品開発のボトルネックになっていた。

そこで、商品開発におけるボトルネック解消のため、スマートフォン用アプリケーションを用いた BLE 通信のテスト自動化を検討し、実現した。本発表では、弊社におけるテスト自動化の実現プロセスと実現方法を紹介し、自動化の取り組みによる定量的な結果と定性的な結果の両方について報告する。

<キーワード>

IoT、BLE、Bluetooth Low Energy、無線、通信、テスト、自動化、スマートフォン、アプリケーション

<想定する聴衆>

ソフトウェアエンジニア、テストエンジニア、プロジェクトマネージャー

<活動時期>

2017 年 1 月～2017 年 5 月

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（ ）

<発表内容>

1.背景

IoT の本格的な普及にともない、通信機能を搭載した様々なデバイスが私たちの生活に欠かせないものになってきている。

弊社では、消費電力の少ない BLE(Bluetooth Low Energy)が主要な無線通信の 1 つになると期待しており、スマートフォン用アプリケーションとの連携を考慮した BLE 対応の商品開発が増えてきている。しかしながら、多種多様な商品へ対応することによって、通信仕様が複雑化し、BLE 通信モジュール開発のテストが肥大化していた。その結果、BLE 通信モジュール開発が商品開発のボトルネックになっていた。

そこで、商品開発のボトルネック解消のため、テストの自動化を検討、実現し、テストの生産性向上を目指した。ただし、自動化することが目的とならないよう、取り組む範囲は費用対効果の最も大きな部分に注力して検討、実践する方針とした。

2.改善したいこと

① 手順が複雑なため、テスト実施に時間がかかる。
② テスト結果の目視確認に時間がかかる。
③ 確認ミスが発生し、やり直しが発生する。
④ 知識のある人しかテストできない。

図 2-1. BLE 通信ログ画面における通信データの一例

3.改善策を導き出した経緯

上記「2. 改善したいこと」の①と②と③については、人が関与する限り改善されない。そのため、人が関与しない方法、つまり自動化が必要であると考えた。

また、上記「2. 改善したいこと」の④についても、自動化を実現することによって、テスターが機械作業的にテストを実施することが可能となるため、自動化によって改善できると考えた。

4.改善策の内容

テスト自動化は、既存のシステム構成を機能拡張することによって実現した。具体的には、アプリケーションの振る舞いをテストスクリプト（Java Script ファイル）に記述し、そのスクリプトをアプリケーションに読み込ませることで、アプリケーションの自動実行を可能とした。

5.改善策の実現方法

自動化の実現のための導入コストを可能な限り低くし、かつ通信テストではスマートフォンとの接続性も検証する必要があったため、

既存のシステム構成拡張によって、自動化を実現した。詳細は下図のとおり。

なお、自動化を実現するにあたって、以下の 2 点を工夫した。

① テストスクリプト作成の難しさを排除するため、テストスクリプト作成まで自動化した。

② テストスクリプトを共通化し、1 つのテストスクリプトで iOS と Android の両方に対応できるようにした。

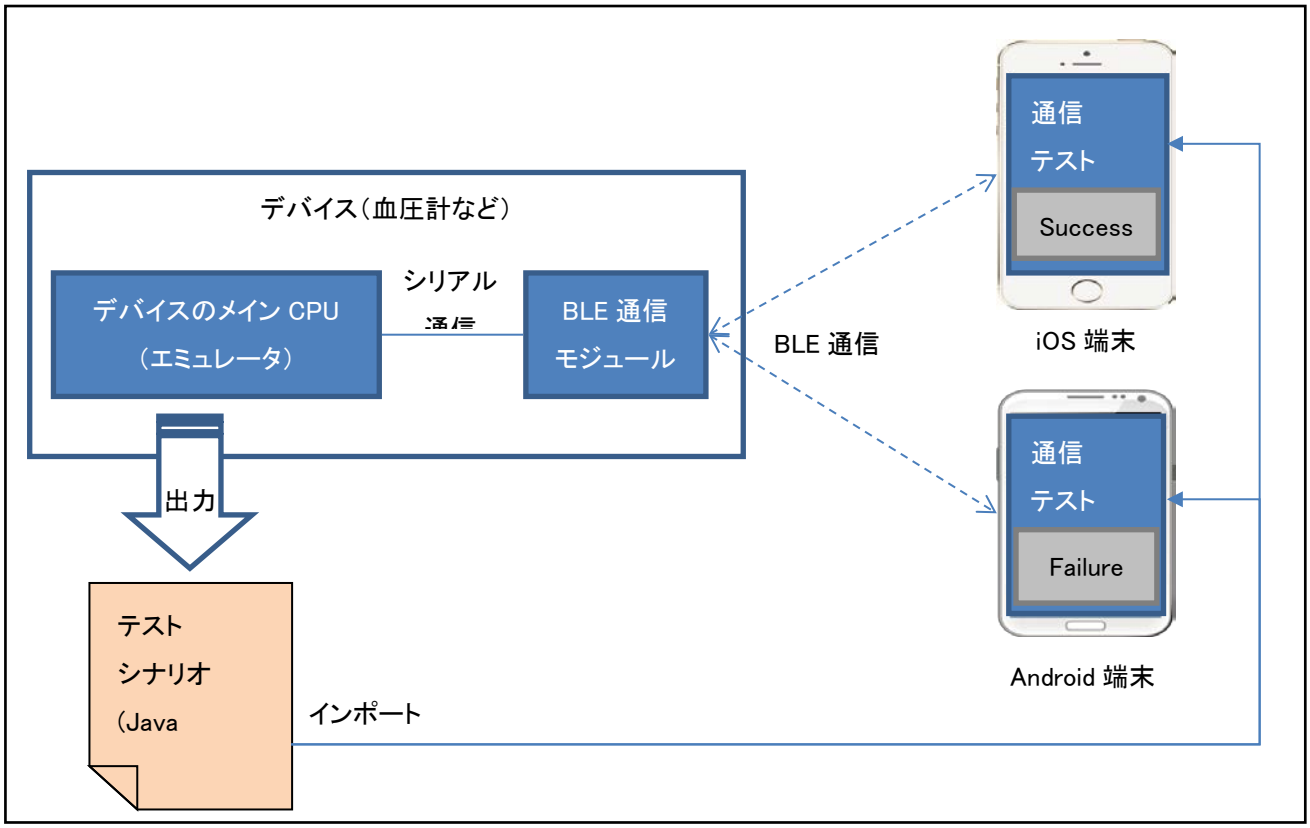


図 5-1. 実現したテスト自動化のシステム構成

＜自動通信テストの手順＞

手順 1：デバイスのメイン CPU のエミュレータを機能拡張し、テストごとのパラメータからテストスクリプトを自動出力する。

手順 2：テストスクリプトをスマートフォン用アプリケーションにインポートし、テストを実行する。

手順 3：テストスクリプトにもとづいて、アプリケーションがデバイスと自動で通信を行う。

手順 4：通信終了後、テストスクリプトにもとづいて、アプリケーションが通信結果を照合し、テスト結果を表示する。

6.改善による変化や効果

＜自動化の実現結果＞

1. 全テストの 21%を自動化することができた。
2. 自動化済みの 21%以外は、費用対効果の観点から自動化を見送った。

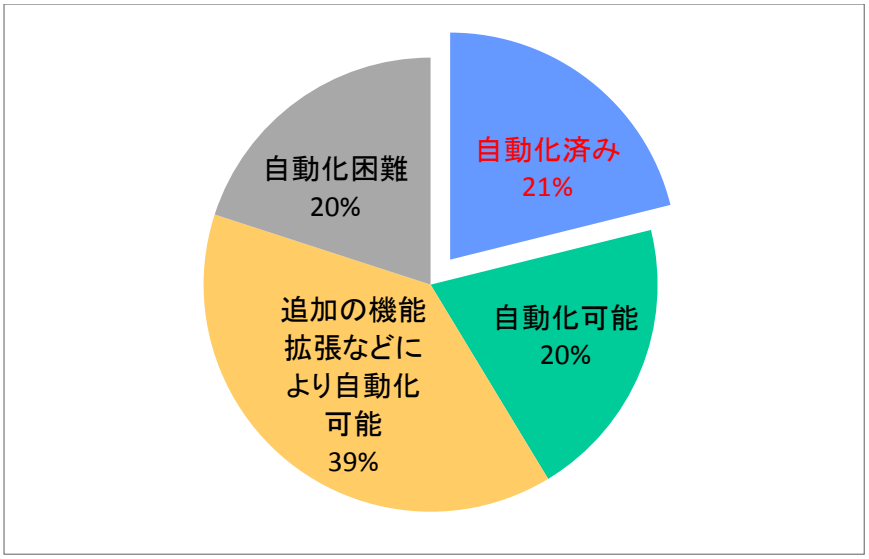


図 6-1. 自動化の実現結果

＜定量的な効果＞

1. 確認ミスによる手戻りを排除することができた。
2. 同テストを手動で実施した場合と比較し、テスト実施の生産性を約 6.5 倍にすることができた。
3. 全テストの 21%を自動化し、全テスト工数の約 13%を削減することができた。

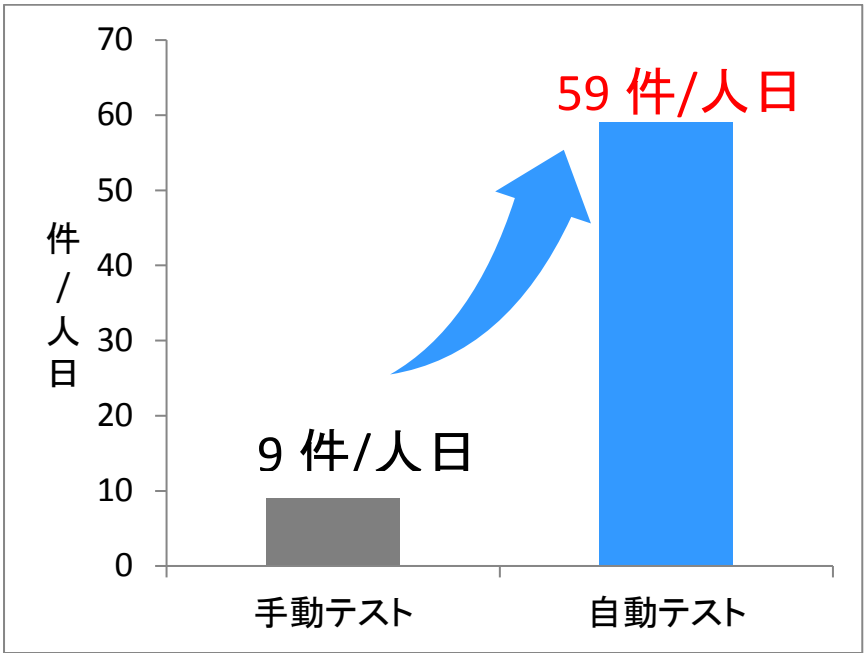


図 6-2.テスト自動化対象のテスト工数の論理値比較

＜定性的な効果＞

1. 簡単にテストできるようになったことで、テスターの肉体的かつ精神的な負担が減った。
2. 誰でもテストできるようになり、テストの計画が立て易くなった。
3. 事前教育などの負担が減り、テスターの追加を前向きに検討可能となった。

7.改善活動の妥当性確認

テスト工数のシミュレーション結果によって、テスト自動化のために費やしたコストを回収できていることが確認でき、本取り組みは妥

当な取り組みであったと考えている。

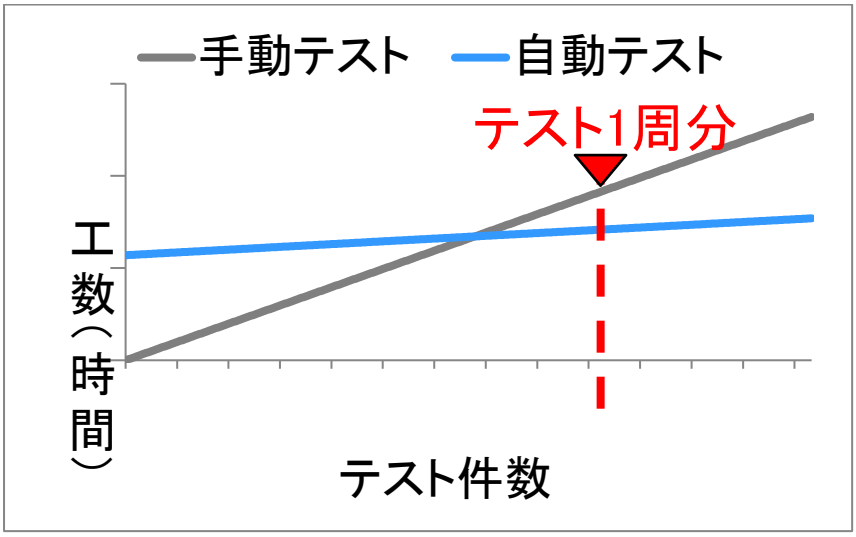


図 7-1. テスト自動化対象のテスト工数の論理値比較

本取り組みでは、全テストの 21%を自動化することができた(図 6-1.自動化の実現結果を参照)が、今後もテストの自動化を推進、拡大し、より効率的にテストを実施していくことが課題である。

A. 参考情報

無し。

2C2「UT 仕様書自動出力システムによる UT 工数削減の取組」野尻優輝 (住友電工情報システム)

<タイトル>

UT 仕様書自動出力システムによる UT 工数削減の取組

<サブタイトル>

<発表者>

氏名（ふりがな）：野尻　優輝（のじり　ゆうき）
所属：　住友電工情報システム株式会社　QCD 改善推進部品質改善推進グループ

<共同執筆者>

氏名（ふりがな）：　服部　悦子（はっとり　えつこ）
所属：　住友電工情報システム株式会社　QCD 改善推進部品質改善推進グループ

<主張したい点>

UT 工数削減の為の取組として、UT 仕様書を自動出力するシステムを作成した。試行した結果、残念ながら工数削減には至らなかったものの、問題点は把握できており、これを解決することで工数削減が実現できると考えている。また、利用者には効果の一部や魅力を感じて貰えており、十分にポテンシャルを秘めた取組といえる。その取組内容と試行の結果を紹介する。

<キーワード>

UT 工数削減、UT 仕様書、自動出力

<想定する聴衆>

UT 設計者、UT 実施者、プロセス改善推進者

＜活動時期＞

2015/1 ～ 2017/2（活動継続中）

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（　　）

<発表内容>

1.背景

弊社にはいくつかのプロセス改善ワーキンググループ(WG)が組織されており品質及びコストの組織目標を達成するために活動を行っている。私の所属する UT 改善 WG では、この目標に対する取組として、これまでにロジック部分の UT に対する自動テストの導入やテストデータ自動生成システムの作成を実施している。[1]

2.改善したいこと

UT 工数を 20%削減する。ただし、改善策の適用後もプログラム品質を維持することと、UT 工程以外の工程に悪影響を及ぼさないことが必須。

3.改善策を導き出した経緯

WG の次の取り組みとして、品質改善を目的に画面に対する UT の設計基準の見直しを実施した時のこと。新基準を用いた UT 設計を試行中、WG メンバーより「1 画面中に POPUP が複数個存在する場合等に同様のテスト内容を複数回記載する作業や、罫線の整形作業等、単純作業や繰り返し作業が多く面倒」「自動化されれば楽だ」という声が上がった。

4.改善策の内容

UT 項目抽出基準と外部仕様書、PG 仕様書を INPUT として UT 仕様書を自動で出力するシステムを作成する。背景で述べた様に、ロジック部分の UT に対しては自動テストの導入を行っている為、画面項目の出力内容や画面遷移の動作等に対する UT 仕様を対象とする。仕様書から画面構成要素等を取得し UT 項目抽出基準に照らして UT 項目を出力する為、同様のテスト内容を記載する作業や罫線等の整形作業といった単純作業・繰り返し作業量が減少する。

5.改善策の実現方法

-1.UT 仕様書 自動出力の前提

弊社では、外部設計書や PG 設計書の情報をリポジトリとして DB に保持し、それを元にプログラムとして動作する開発フレームワークを利用している。(図 1)

よって、機能ごとにどのような画面があり、どのような項目が表示されるのかといった情報が DB から取得できる状態になっている。(エラーチェックやビジネスロジックなどは別途 java で作成する。)

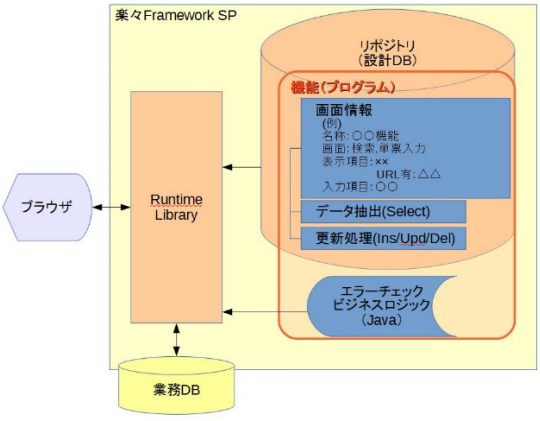


図 1 フレームワーク概要

-2.UT 仕様書 自動出力の仕組み

UT 仕様書を自動出力するにあたり、画面要素や処理内容といった検証対象毎に何をテストすべきかを"UT 項目抽出基準"として定義した。(図 2)

			全社標準UT項目		テンプレート			
No	分類	分類名	番号	名称	テンプレートNo	検証項目	テストケース	テスト項目
1	U0010	表示データ加工-画面表示-画面-サブメニュー	10	サブメニューの加工	100	サブメニューの加工／活性・非活性	%userspec%	メニュー状態「%menustatusnm%」が画面に適用されていることを確認する %status%
2	U0010	表示データ加工-画面表示-画面-サブメニュー	20	画面遷移	101	画面遷移(カスタマイズあり)	%userspec%	
3	U0020	表示データ加工-画面表示-画面-表示用部品-単票	30	表示件数(表示用部品・単票)	102	表示件数	テストデータAを準備する。 検索画面で条件Bを入力し検索ボタンを押下する	1件表示されること
4	U0030	表示データ加工-画面表示-画面-表示	40	表示件数(表示用部品・一覧)	103	表示件数	テストデータAを準備する。 検索画面で条件Bを入力し検索ボタ	0行表示されること

図 2　UT 項目抽出基準(抜粋)

リポジトリに格納された画面の種類、表示項目等の情報を UT 項目抽出基準と照らし合わせ、テスト仕様を生成する。(図 3)

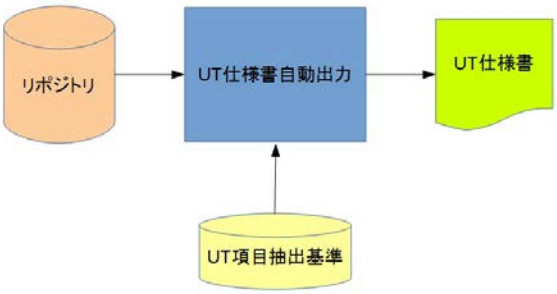


図 3　仕組み概略図

図 4 は一般的な照会機能の画面遷移を表している。この画面には、氏名、品目に対してリンクがあり、そのリンクをクリックするとユーザー情報、品目情報を参照できるようになっている。

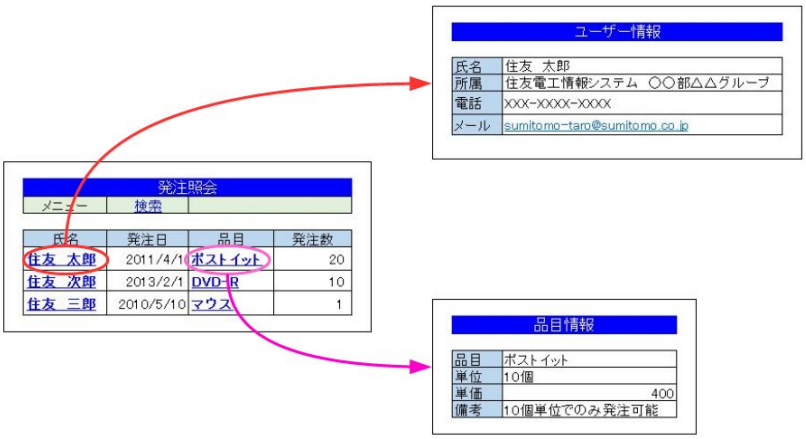


図 4　画面イメージ

このリンクに関するテスト仕様生成のイメージを図 5 に示す。

まず、リポジトリよりリンク有の項目を抽出する。次に、UT 項目抽出基準よりテスト仕様(テストケース、テスト項目)を抽出する。テスト仕様の文言の一部をリポジトリから取得した項目名や遷移先の情報で置き換え、UT 仕様とする。

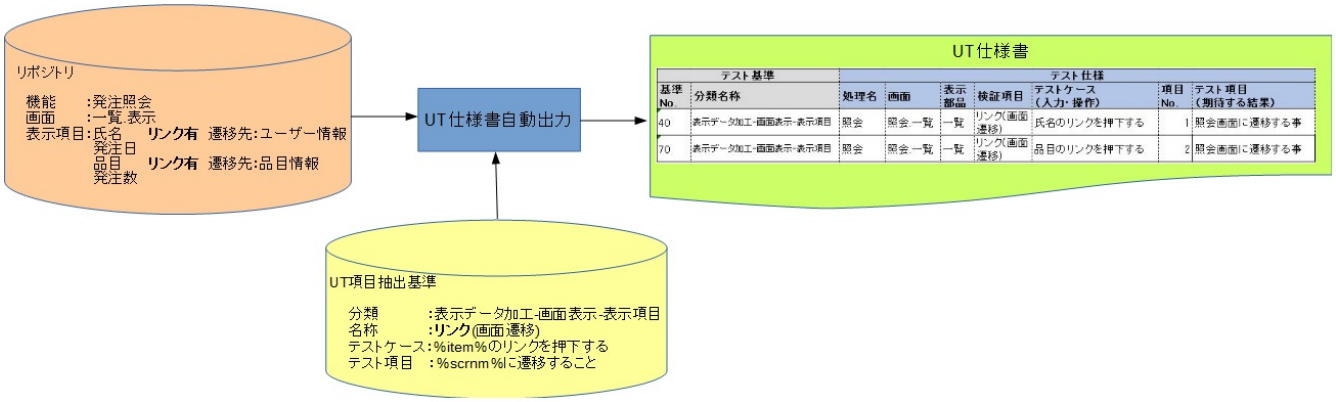


図 5　テスト仕様書生成イメージ

6.改善による変化や効果

試行 PJ を実施し、以下の評価を行った。

-1.UT 工数削減

改善策適用前後の UT 工数と、その内訳である UT 設計工数、UT 実施工数を比較した。(表 1)

改善策適用後の UT 工数は、適用前の約 1.5 倍と増加してしまった。内訳を確認すると、UT 設計工数が約 3 倍に増加している。

表 1　改善策適用前後の UT 工数

	適用前	適用後	変化
UT 工数	100	145	50%増
内　UT 設計	38	114	200%増
内　UT 実施	62	31	50%減

※適用前 UT 工数を 100 とした相対値

-2.プログラムの品質維持

改善策適用前後のプログラム規模当りの不具合数を比較して評価を行った。

適用後 PJ はサンプル数が少ないが、プログラム規模当りの不具合数の平均は上がっておらず、品質が維持できていることが確認できた。集計結果を示す。(図 6)

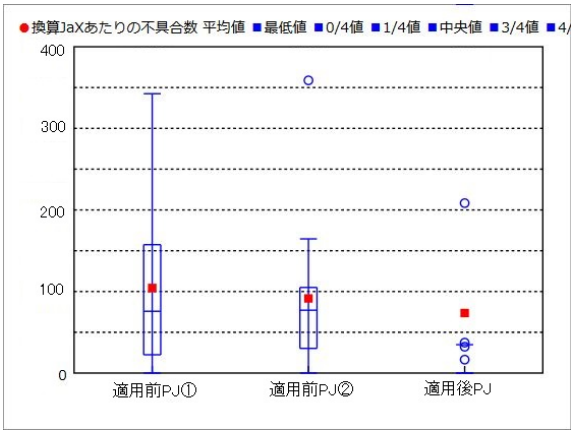


図 6　プログラム規模あたりの不具合数

※適用前 PJ①の不具合数の平均を 100 とした相対値

-3.UT 工程以外への悪影響の有無

特別な準備、想定外の作業が発生しなかった為、UT 工程以外への悪影響はなかったと判断した。

-4.ツールの魅力性

本ツールを今後も利用したいか、試行メンバーにアンケートを実施した。結果、”本ツールを今後も利用したい。ポップアップウィンドウの動作など、自動出力されて嬉しいものはあった。”との回答が得られた為、魅力ありと判断した。

7.改善活動の妥当性確認

本施策は継続し、改善と試行のサイクルを回していくべきと考える。残念ながら当初の目標である"UT 工数を 20%削減"は達成されず、むしろ工数が増加する結果となった。しかし、自動出力によって単純作業・繰り返し作業が少なくなっていることを感じてもらえている。UT 設計工数の増加要因を解消すれば UT 工数削減効果が得られると考えている。

8.原因分析

UT 仕様書自動出力後の作業手順を見直すと、自動出力 UT 仕様書の編集に時間が掛かっていると考えられる為、変更、削除、追加件数を調査した。その結果、“削除された自動出力した UT 項目“と、”追加された UT 項目“が多いことがわかった。つまり、”自動生成 UT 項目に不要なものが多い“、”自動出力できる UT 項目がまだある“ということである。また、削除件数が想定よりも大量であった為、UT 仕様書自動出力導入後に増加した工程（”自動出力された項目の内容理解＋削除判定“）によって、UT 仕様書の工数が増加したとも考えられる。

9.改善検討

編集作業減少を目指し、WG 内で自動生成 UT 項目の出力内容の見直しを行っていく。

A. 参考情報

[1]服部悦子, “テストデータ自動生成による品質・コストの改善”, SPI Japan 2014,2014

2C3「システムテスト自動化の普及・展開に向けた取り組み」小笠原秀人 (東芝)

<タイトル>

システムテスト自動化の普及・展開に向けた取り組み

<発表者>

氏名（ふりがな）：小笠原秀人（おがさわらひでと）
所属： 株式会社東芝

<共同執筆者>

氏名（ふりがな）： 中野隆司（なかのたかし）
所属： 株式会社東芝

<主張したい点>

システムテストの自動化は、過去にも多くの部門で導入を試みてきたが、定着し運用できている部門は少ない。本発表では、システムテスト自動化導入の阻害要因を分析し、その阻害要因を解決するための施策を立案し、実践している内容を紹介する。

<キーワード>

システムテスト自動化、テスト自動化ツール、テスト管理、テスト開発者、テストスクリプト

<想定する聴衆>

システムテストの品質と効率を高めたいと考えている部門や推進者
テスト自動化ツールの導入を計画している部門や推進者

＜活動時期＞

2016 年 7 月～

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
■ 改善活動を実施したが、結果はまだ明確ではない段階
□ 改善活動の結果が明確になっている段階
□ その他（ ）

<発表内容>

1.背景

ソフトウェア開発のテスト期間とテスト工数の比率は約 40%に達すると言われている（※1）。テスト期間を短縮し、テスト工数を削減するためには、上流工程から品質を作り込むことが重要であり、品質作り込みのための改善を含めた開発を優先すべきである。しかし、現実的には、製品リリースの時期が決まっているために現状の開発方法で進めなくてはいけない、改善を含めた開発のためにリソースが十分に確保できないといった理由で、実現できていないことが多い。一方、システムに求められる品質要求は高くなり、かつ、システムが利用される環境の多様化により多量のテストを短期間に行う必要あり、テストの質と量に対する要求は高まっている。

※1 ：ソフトウェア開発データ白書 2016-2017, IPAS/SEC, 2016/10.

2.改善したいこと

開発している製品はシリーズ製品が多く、ある母体をベースにした派生開発である。派生開発が中心となる開発状況において、テスト期間とテスト工数の増大は、廉価版製品開発などのリリースが計画どおりに行えないといった問題につながってしまう。そこで、テスト工数を大きく削減し、製品バリエーションに対応した最適な開発を実現することが求められている。

しかしながら、社内における派生開発においては、初期開発に比べて開発工数は削減できるが、テスト工数は大きく削減できないことが多い。その理由は大きく二つあり、一つは、製品としての品質を担保するためには変更の影響が少ないと思われる部分に対してもテストを行うが、この部分は人手に頼ることが多く、ある一定以上のテスト工数が必要となるためである。もう一つは、テスト自体の品質やバリエーションの不足により、テストの抜けもれが生じてしまい、後戻り工数が増大してしまうためである。派生開発における製品バリエーションに対応した最適な開発を実現するためには、次の 2 つの課題を解決する必要がある。

課題①：繰り返し行うテストによる工数の増大

課題②：テストの抜けもれによる後戻り工数の増大

3.改善策を導き出した経緯

2 章で示した課題①、課題②を解決するために、東芝グループ内における多くの部門では、過去にシステムテストの自動化に取り組んだ経験を持っている。しかしながら、システムテストの自動化が定着し、効果的・効率的に運用できている部門は少ない。

一方、テスト自動化を促進するためのツールという観点で見ると、近年、多くのテスト自動化のためのツールが利用できる状態になっている（市販ソフト、フリーソフトの両方ともに（※2））。そこで、テスト自動化のためのツールが活用できるのを好機と捉え、東芝グループ内にシステムテストの自動化を推進し、課題①、課題②の解決を目指すことにした。次に、システムテストの実施状況を見てみると、初版開発時に作成したシステムテスト仕様書をベースに、さまざまな観点をある程度網羅したシステムテスト仕様書が整備され活用されていることが多かった。新しい施策を展開するための負荷をできるだけ少なくするために、改善策は、次の 2 つのステップに分けて実施することにした。

ステップ 1 の改善策：自動化で工数を削減する（課題①の解決）
既存のテスト仕様をベースにテストスクリプトを作成し適用し、その後、複数の製品シリーズに展開して、テスト工数を大幅に削減する。

ステップ 2 の改善策：抜けもれないテストスクリプトを作る（課題②の解決）
抜けもれないテストスクリプトを効率良く開発・維持し、QCD を満たす開発を実現する。

図 1 に、システムテスト自動化プロセスと上記ステップの関係を示す。

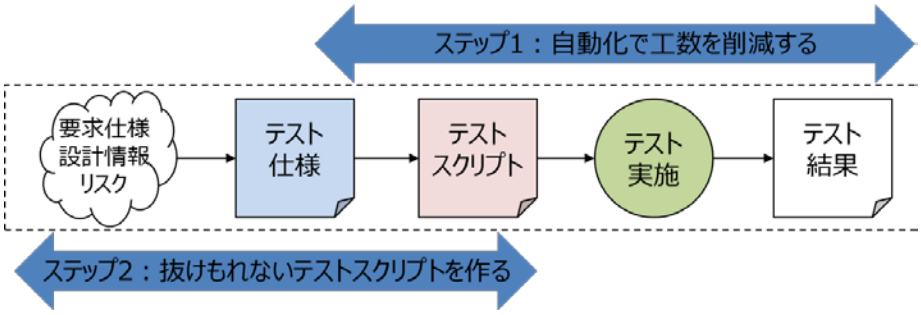


図 1　システムテスト自動化プロセスと改善策の実施順番

※2：市販では、Ranorex、TestComplete、フリーソフトでは、Selenium、Sikuri、UWSC などがある。

4.改善策の内容

3 章で示したステップ 1、ステップ 2 の改善策を実施するための阻害要因を分析した。その結果を表 1 に示す。

表 1　ステップ 1、ステップ 2 の改善策を実施する際の阻害要因

ステップ	改善策	阻害要因	阻害要因が導かれた過去の経験
ステップ 1	自動化で工数を削減する	導入・維持コストがかかり効果が出ない	導入コストが高いため手作業の方が早いと思われる
			テストスクリプトの保守コストが高い
			自動化ツールを使いこなすまでに数ヶ月以上かかる
			テストスクリプトの変更割合が多く、管理が難しい
ステップ 2	抜けもれのないテストスクリプトを作る	テスト仕様に抜け漏れがあり効率的・効果的に作れない	自動化しても不具合の発見に繋がらない

次に、阻害要因に対する施策と施策内容を検討した。表 2 にそれぞれの阻害要因に対する施策と施策内容を示す。ここで挙げた施策を東芝グループ内の複数のプロジェクトに適用し、全社的に活用できるガイドやツール、方法論などを整備する計画を立案した。

表 2　阻害要因に対する施策と施策内容（システムテスト自動化の普及・展開のための全社施策）

阻害要因	施策	施策内容
ステップ 1 の課題：導入・維持コストがかかり効果が出ない（2016～2018）	①　導入障壁の克服	ツール選定、自動化範囲の見極め、自動化の計画立案と実行管理の支援
	②　体制を整備	開発担当者、支援担当者、テスト開発者の役割とプロセスを明確にする
	③　環境を整備	仕様、テストスクリプト、実行結果の一元管理できる環境を構築する
	④　再利用でき保守しやすいテストスクリプトの開発	テストスクリプト開発技術の開発
ステップ 2 の課題：価値の高いテストスクリプトを効果的に作れない（2017～2020）	⑤　テスト設計の品質向上のための診断と改善	テストプロセス診断・改善技術（テストプロセス成熟度モデル TPI Next の利用）
	⑥　テスト仕様からテストスクリプトを効率よく作る	テストスクリプト自動生成技術の開発

5.改善策の実現方法

表 2 に示した全社施策の最初の適用プロジェクトとして、社会インフラ系の大規模システムの開発を対象に、システムテスト自動化に取り組んだ。適用期間は、2016 年 10 月から 2017 年 3 月までの半年間である。この開発は、製品系列シリーズの最初の開発であり、この開発以降数年間に渡り、シリーズ製品を複数の場所に設置することが決まっている。シリーズ製品は、ハードウェア構成には大きな変更がなく、個別の設置場所毎に小規模な仕様変更が入る予定である。

この開発においては、表 2 の施策①～④を実践した。これらの施策を実践するために、開発担当者、支援担当者、テスト開発者の 3 つの役割を定義し、リソースを割り当てた。テスト開発者はベトナムの技術者である。施策①～④の内容を以下に示す。

- 施策①：導入障壁の克服
  - テスト自動化ツールとして Ranorex（市販ツール）と UWSC（フリーソフト）を比較し、Ranorex を選定。
  - 開発担当者と支援担当者でシステムテスト自動化の範囲を定め、自動化の戦略と計画を策定した。
  - 作業項目や Q&A は、Redmine で管理。テスト仕様に関する質問や確認は、テスト開発者から開発担当者にチケットとして発行した。
  - テスト開発者が、日本の事業所に約 1 ヶ月間駐在し、オンサイトでテストスクリプトを開発するとともに、テスト実行環境を整備した。開発者 2 名と通訳 1 名が来日した。
  - 実践結果からノウハウや留意点を洗い出し、「システムテスト自動化ガイド」を作成した。

- 施策②：体制を整備
  - ベトナムのソフトウェア開発の拠点に、約 10 名で構成されるテストセンターを設置（テストセンターでは、本開発以外にも複数のテスト自動化プロジェクトを担当している）。
    - テストセンターのメンバには、開発プロジェクトにおけるテスト自動化の目的や目標、社内におけるテスト自動化推進の将来像などを説明し、テスト自動化が期待されている技術であることを伝えている。
  - 毎週、開発担当者と支援担当者、支援担当者とテスト開発者の 2 つの定例会を開催。定例会での主要な確認／検討項目は以下のとおりである：
    - テストスクリプトの開発状況の確認
    - テスト自動化の範囲の見極め
    - テストスクリプト開発や実行に関するリスクの先取り
    - 自動化されたテストの実施結果の確認
    - 不具合と思われる現象の確認

- 施策③：環境を整備
  - テストセンターと連携して、テストスクリプトとテスト仕様の一貫性を維持するためにテスト管理ツール TETRAPLUS（※3）を軸としたテスト自動化環境を整備した。

- 施策④：再利用でき保守しやすいテストスクリプトの開発
  - 実際にテストスクリプトを開発し運用しながら検討した。テストを効率よく実行するための前処理、後処理や共通処理などを切り出し、主にテスト環境やテストデータの変更に対応しやすい構造にした。

※3：支援担当部門が開発し、東芝グループ内で広く活用されている。テスト仕様とテスト結果を一元管理できる Web システム。テスト仕様とテスト結果は、Excel を介してエクスポート／インポートできる仕組みを持つ。

6.改善による変化や効果

施策①～③の実践による結果と効果を以下に示す。

- 「施策①：導入障壁の克服」の結果
  - 自動化の向き不向き、再利用を考慮してテスト自動化対象範囲を特定した（2,500 件のテスト仕様のうち、約50%の自動化が有効と判断）
  - 自動化対象の約 1,200 件のテスト仕様に対応するテストスクリプト 260 個の開発を完了した
    - 本システム開発の場合、現状のテスト仕様の書き方だと、テスト仕様 5 件に対してテストスクリプト 1 個を作成するという割合になる

- 「施策②：体制を整備」の効果
  - 自動化の導入により次機種以降、テストコストを 30%削減できると見積もっている（開発担当者による）
    - 次期種で初期導入コストを回収できる見込み
  - テストセンターの活用により、前機種と比較してほぼ同等のコストでテスト自動化部分を含めたテストを実施できた（図1）。さらに、自部門でテスト自動化する場合に比べて、テストセンターの利用により初期導入コストを 40%削減した（図 2）。

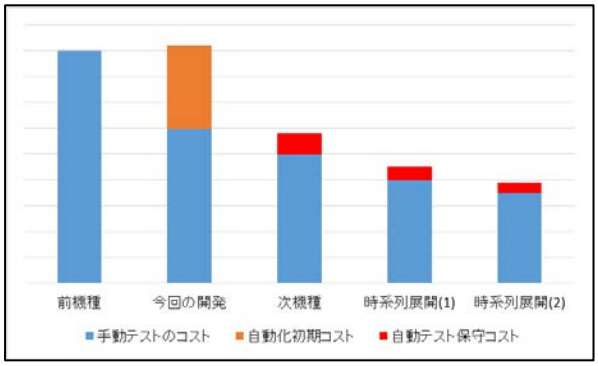


図 1　システムテストのコスト推移予測

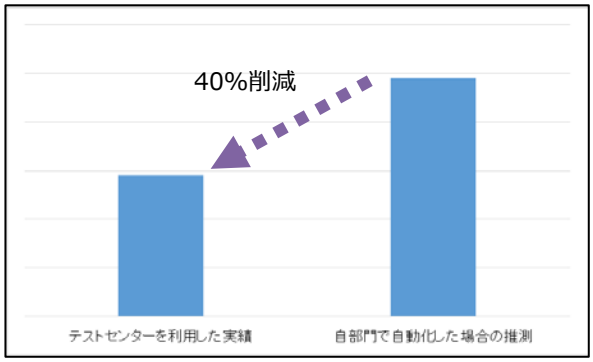


図 2　初期導入コストの評価

  - 開発者／担当者が担うべき役割を明確にして実践した結果（図 3）、予定した期間で予定した範囲のテスト自動化が実現でき（施策①）、導入コストも開発部門に大きな負担にはならなかった（施策②）。これは、開発担当者、支援担当者、テスト開発者による構成が有効に機能した結果と考えている。

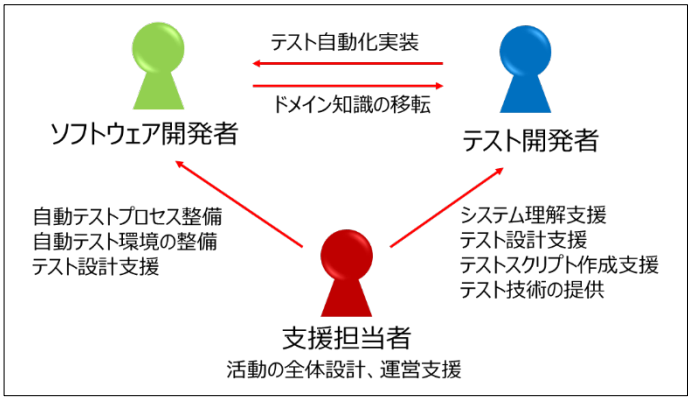


図 3　各開発者／担当者の役割

- 「施策③：環境を整備」の効果
  - テスト仕様、テストスクリプト、テスト結果を一元管理する仕組みを確立し、それを実現する環境を構築できた。これにより、テスト結果の可視性が向上した。さらに、既存のテスト仕様を利用できることによる導入コストの低減がはかれた。また、既存のテスト仕様をベースにテストスクリプトを開発することの課題も見えてきた（何が正解か分かりづらい、仕様に対する網羅性が低い、など）。この課題は、「ステップ２の課題：価値の高いテストスクリプトを効果的に作れない」における「施策⑤：テスト設計の品質向上のための診断と改善」、「施策⑥：テスト仕様からテストスクリプトを効率よく作る」のインプットとなる。
  - 構築したテスト自動化のための環境を図４に示す。

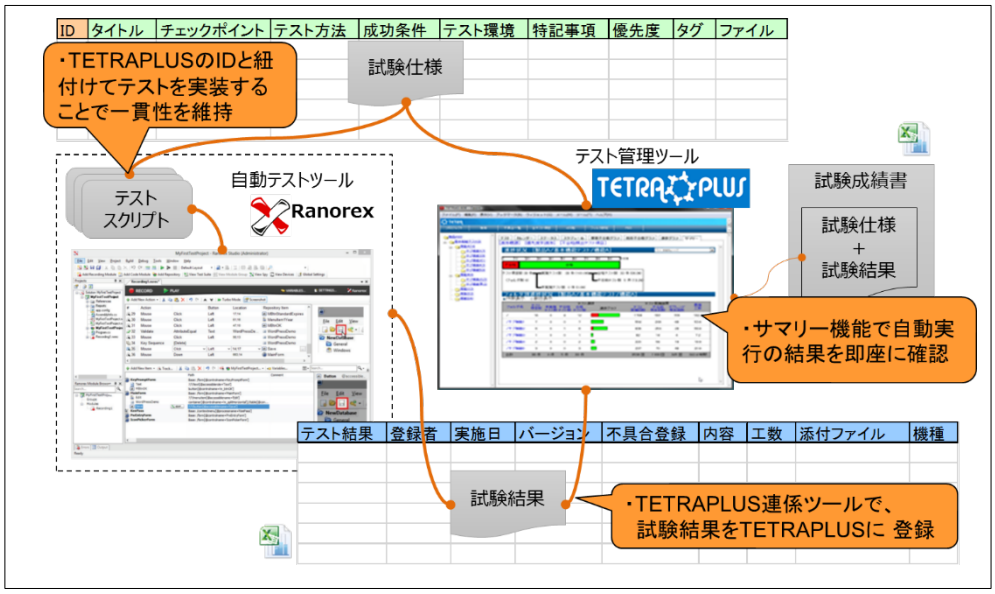


図４　構築したテスト自動化のための環境

7.改善活動の妥当性確認

テストスクリプト完了後、システムテストの期間中（約３週間）に、毎晩自動テストを実行した。その期間も、仕様変更や不具合対応でソースコードも更新されていた。自動テストの実行結果、不合格となったテストが検出されたが、それは機器の設定漏れのミスであった。また、不具合として ３ 件を検出したが、テスト結果を検証した結果、不具合でないこと確認した（検証方法が不十分であった）。それ以外は、すべて合格であった。

通常、自動テストで実施したテストを実行するには、一人で １ 週間程度の時間をかけて実施していたが、約 ３ 時間で実施できるようになった。このように、テスト実施時間を大幅に削減できたことによって、変更に対するデグレードがないことを翌日には確認できるようになった。プロジェクト側からは、次機種の開発以降も、テストの自動化の取り組みを継続するという評価を得た。

テスト自動化の取り組みをとおして得られた課題と知見は以下のとおりである。これらの課題は、よく考えれば当然出てくる内容であるが、実際に、大規模システムの実機を利用したテスト自動化を実践したことで、始めて実感として気がつくことができた。

- 課題(1)：大規模システムにおける自動化
  - 複数の機器やサブシステムの影響をうけて自動テストが安定しないことがある
    - ✧　得られた知見：外部条件を付加したテストスクリプトの開発が必要である

- 課題(2)：判定結果の見極め
  - 不具合なのか、実行環境の問題なのか識別に時間がかかる
    - ✧　得られた知見：自動実行結果に基づいて効率的に原因分析する仕組みが必要である

- 課題(3)：ヒューマンファクターによる自動化の中断
  - 自動実行中に、機器の切断や開発・デバックなどによる使用によって、自動化が中断した
    - 得られた知見：自動テストを開発プロセスに組込み、周知徹底することが重要である

今回のテスト自動化の対象であるプロジェクトが、次機種以降もテスト自動化の取り組みを推進するという判断をしたこと、テスト自動化を推進する際の課題を明らかにできたことから、本改善活動は妥当だと考えている。今後、今回得られた課題と知見を活用し、テスト自動化の取り組みをより推進していく。

A. 参考情報

[1] システムテスト自動化 標準ガイド: Mark Fewster, Dorothy Graham 著, テスト自動化研究会 訳, 翔泳社, 2014/12.

2D1「社内エキスパートの育成による、ソフトウェアプロダクトライン(Software Product Lines)の全社展開」丹羽徹（オムロン）

<タイトル>

社内エキスパートの育成による、ソフトウェアプロダクトライン(Software Product Lines)の全社展開

<サブタイトル>

（なし）

<発表者>

氏名（ふりがな）：丹羽 徹（にわ とおる）
所属： オムロン株式会社 グローバルものづくり革新本部 開発プロセス革新センタ SPILIT 推進部

<共同執筆者>

氏名（ふりがな）：赤松 康至（あかまつ やすゆき）
所属： オムロン株式会社 インダストリアルオートメーションビジネスカンパニー 商品事業本部
技術開発センタ 第 3 技術部

<主張したい点>

ソフトウェアプロセス改善（SPI）を推進する中で、特に新規商品・リニューアル商品に対してはソフトウェアプロダクトライン（SPL）の考え方を用いたプロジェクトをタイミングよく立ち上げて、開発現場に適用することが効果的であること。タイミング良く効果的に SPL を適用するためには、製品開発事業横断的に SPL のエキスパートを育成し、プールしておくことが有効であること。

<キーワード>

ソフトウェアプロダクトライン、SPL、ドメイン分析、コア資産、ソフトウェアプロセス改善、SPI、大規模化、人材育成、エキスパート、全社展開

<想定する聴衆>

ソフトウェアの大規模化に悩む経営層、プロジェクトリーダ層、アーキテクト、SEPG

<活動時期>

2011 年 5 月～

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（ ）

<発表内容>

1.背景

1933 年に創業し、スイッチ・リレーからスタートした弊社事業も、今や FA 事業、ヘルスケア事業、公共事業、車載事業など多角化が進んでいる。また、その中でほとんどの商品にマイクロコントローラを搭載し、組込製品としてソフトウェアで機能を実現することにより、ソフトウェアが製品の価値の大部分を占めるようになってきている。さらに、ますます高まる製品の高機能化、ローコスト化（ハードをソフトで実現）のニーズに伴い、製品に搭載するソフトウェアの規模は増える一方である。

このような中、弊社ではソフトウェアの QCD 向上を狙いにソフトウェアプロセス改善活動（SPI）に取り組んで来ており、プロセスリファレンスモデル（CMMI など）を用いた改善活動が、複数の事業体である各ビジネスカンパニーで実施している。一方で、弊社の製品の特性上、商品群ごとに 5～10 年スパンでハードウェアも含め新規開発もしくは大きく製品リニューアルが行われ、その後複数の派生開発を行うというサイクルを繰り返しており、この新規・リニューアルでのソフトウェアの作り方がまずいと、後の派生開発の QCD に大きく影響を与える構造になっている。例えば派生開発でマネジメントから「なぜこんなに開発費がかかるんだ？」とか、「なぜそれくらいの変更を最初から想定しておかなかったのか？」とか、「ソフトがボトルネックで新たな派生開発を受注できない」とか言われることが多い。これまでの SPI はどちらかというと短期で複数のプロジェクトが走る派生製品開発の部分をターゲットとしており、製品の新規開発やリニューアルは各製品群で設計者が自己流で進めることが多かった。

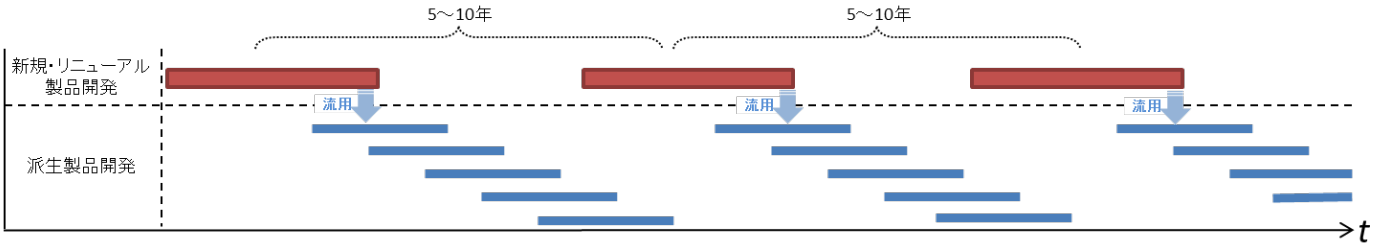


図 1.新規・リニューアル製品開発と派生製品開発のタイムスパン（イメージ）

2.改善したいこと

全社として商品群ごとに 5～10 年スパンで来る新規・リニューアル製品でのソフトの作り方を改善し、その後の派生開発も含めトータルの QCD をより大きく改善したい。特に昨今のソフトの大規模化と市場の QCD 要求の高まりにより、事業としてより大きな改善が期待されているためである。

3.改善策を導き出した経緯

派生開発も含めて製品群トータルで QCD 改善をする手法としてはソフトウェアプロダクトライン（SPL）が有名だが、プロセスリファレンスモデルではほとんど触れられておらず、組織プロセスとしてもほとんど定義されていいなかった。また仮に SPL の手法を組織プロセスとして定義し人材を育成しても、開発現場では 5～10 年に 1 度程度しか使われないため、プロセスは陳腐化し人材は配置転換となり、組織としての知見が蓄積できない状態であった。

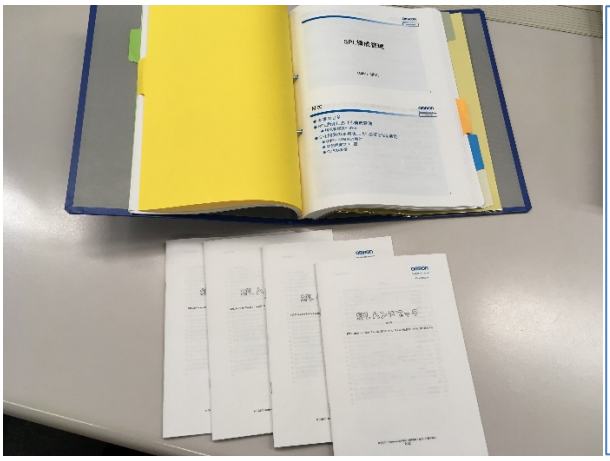
4.改善策の内容

そこで、SPL の手法を組織プロセスとして定義することはやめ、ビジネスカンパニー横断的に SPL を指導できるエキスパートの人材育成に注力することとした。本社スタッフとして人材をプールし、エキスパート人材として複数のビジネスカンパニーを渡り歩くことで、より多くの SPL 経験をコーポレートとして特定のスタッフに短期間で蓄積できるようにし、SPL の全社展開を図った。

5.改善策の実現方法

SPL エキスパートの実現にあたっては、各事業ライン・スタッフ部門等から適性を見極めた上で採用を行い、本社スタッフとして SPL エキスパートチームを構成。その上で SPL エキスパートチーム内での能力向上、知見共有のための資料である”マテリアル”を作成し維持。また、エキスパートとして必要な能力を示した”スキルマップ”を作成し、Off-JT/OJT を通じて体系的に能力を向上

できるようにした。



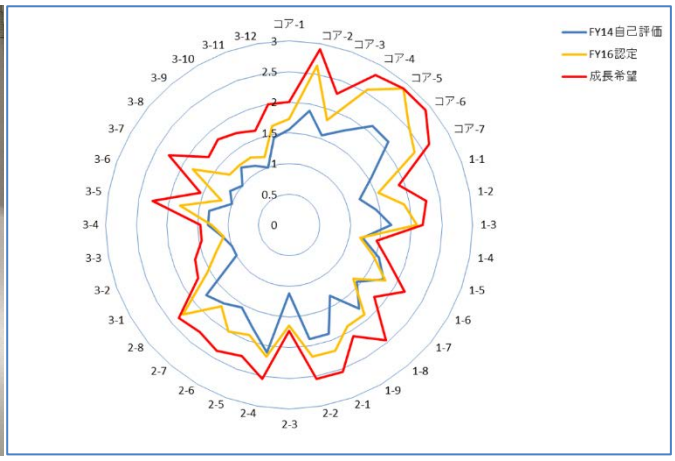


図 2.マテリアル(エキスパート育成用[約 400 頁]:上)と 図 3.スキルマップに基づくエキスパートのスキルアップの推移
SPL ハンドブック(SPL 紹介用冊子:下)

6.改善による変化や効果

結果として SPL エキスパートが関与した複数のビジネスカンパニーの製品群開発で大きな効果が出た。具体的には SPL エキスパートがビジネスカンパニーの製品群開発に企画段階から入り込み、SPL を適用すべきかどうかの ROI 判断、ドメイン分析の方法、コア資産の作成方法、コア資産の維持の方法などを指導。

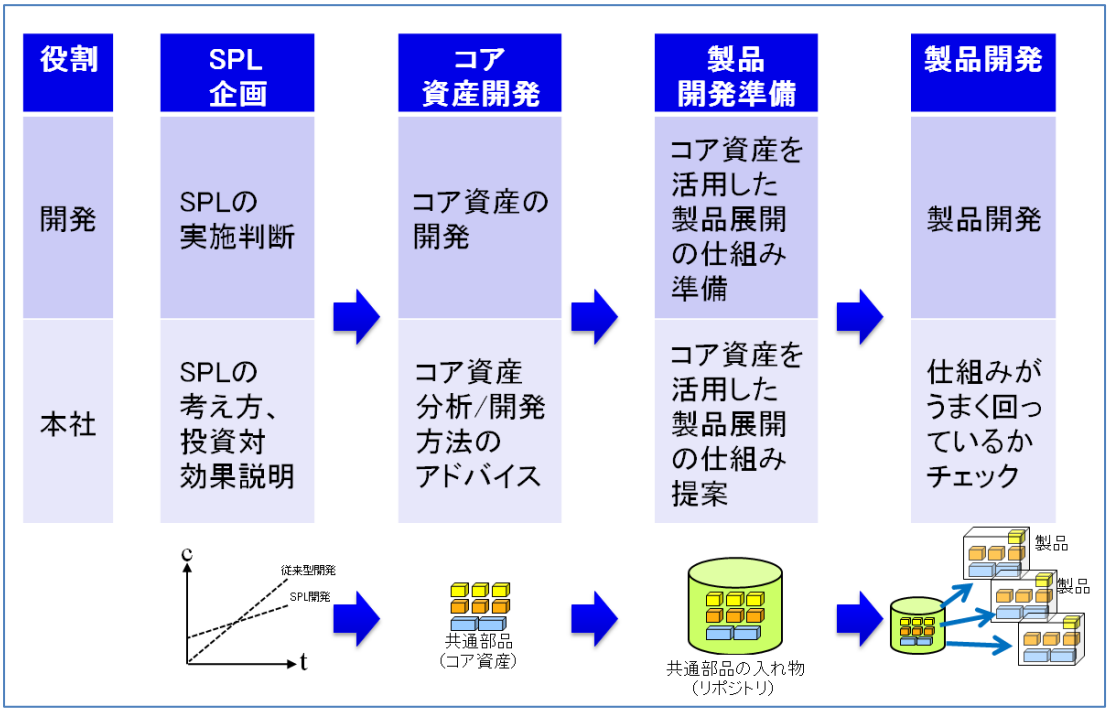


図 4.活動のステップと本社エキスパートと開発現場との役割分担

その結果、例えばコア資産開発での早期の客先デモによる数十億円規模の商談獲得、派生開発での開発費従来比 30%減、ソフトウェアテストの工数従来比 40%減（見込み）、従来は製品ごとに 6 種類あったソフトを 1 つにすることで生産ラインの段取り大幅改善、などが成果として出た

7.改善活動の妥当性確認

開発現場では、どうしても目の前の課題に目が行きがちである。そこに訓練を受け長期的な視点を持った別組織からの人材(SPL エキスパート)が入ることで、短期と長期のバランスを取る打ち手が実施できたと考える。これまで世の中に SPL という優れた

手法があるにも関わらず、なかなか現場には導入できないという歯がゆい思いを本取り組みによって解消することができた。

ただし、実際に SPL を現場に取り入れるにあたっては、別組織からのエキスパート人材が開発現場の中で関係性を保ち、共通のゴールを目指して活動を実施することは並大抵ではなかった。最初はエキスパートとして現場に入っても全く相手にされなかった。現場では SPL 以外にもっとさまざまな問題で悩んでいたからである。そこで最初はそれぞれの開発現場が困っている直近の課題解決も同時に行った。その中で少しずつ現場との信頼関係が構築され、エキスパートによる長期的な打ち手にも耳を傾けてくれるようになり、開発現場と共に大きな打ち手、大きな改善につなげることできた。

また、本取り組みは弊社の経営層からの評価も高く、本社としての活動継続も承認された。エキスパートチームの更なるスキルアップを図り、より大きな成果を確実に出せるよう努めていく。

本知見が同じようにソフトの大規模化に悩まれる経営層、プロジェクトリーダ層等の参考になれば幸いである。

A. 参考情報

[1] Paul Clements and Linda Northrop, “Software Product Lines: Practices and Patterns”, Addison-Wesley（2001）

[2] Paul Clements and Linda Northrop, “A Framework for Software Product Line Practice, Version 5.0”, http://www.sei.cmu.edu/productlines/frame_report/index.html

[3] 今関 剛, “組込みプレス Vol.4「戦略的再利用のススメ／プロダクトラインの歩き方」”, 技術評論社（2006）

[4] ISO/IEC 26550: 2013, Software and systems engineering -- Reference model for product line engineering and management

[5] Robert McFeeley, “IDEAL: A User’s Guide for Software Process Improvement”, Software Engineering Institute（1996）

[6] CMMI V1.3 翻訳研究会, “開発のための CMMI® 1.3 版”, JASPIC（2010）

[7] “Standard CMMI® Appraisal Method for Process Improvement, Version 1.3b: Method Definition Document for SCAMPI A, B, and C”, CMMI Institute（2014）

2D2「プロダクトライン開発における複数製品導出の同時並行開発方法の提案」荒木邦彦（デンソー）

<タイトル>

プロダクトライン開発における複数製品導出の同時並行開発方法の提案

<サブタイトル>

衝突軽減システム搭載車種の拡大に向けて

<発表者>

氏名（ふりがな）：荒木邦彦（あらきくにひこ）
所属：㈱デンソー

<共同執筆者>

氏名（ふりがな）：林健吾（はやしけんご）
所属：㈱デンソー

<主張したい点>

安全システムへの要望が高まり、超音波を利用した衝突軽減システムの搭載車種が拡大している。我々はプロダクトライン開発で工数の低減に取り組んでいるが、納入時期が近接する同系列 3 車種 10 品種の開発受注に対して、開発リソースが不足する問題が生じた。そこで、これらの開発を単一のプロジェクトにまとめて一本のパイプラインとしてコントロールする開発方法を考案した。納入日に対して開発活動の順序を最適化することで、3 車種の納期を達成した。本開発方法の効果を報告し、実践から得られた適用条件と改善の余地について報告する。

<キーワード>

プロダクトライン開発、SPLE、自動車システム

<想定する聴衆>

ソフトウェアエンジニア、プロダクトライン開発者、組込系技術者、SEPG

＜活動時期＞

2016 年 10 月～2017 年 2 月

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動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、自動車におけるアクセルとブレーキの踏み間違い事故を減らすため、超音波センサーを用いた衝突軽減システムの開発を進めている。「このシステムにより踏み間違え事故を約 7 割低減できている」という市場での調査結果もあることから、このシステムは踏み間違い事故に対して非常に有用であるといえる。

連日のニュースで取り上げられる高齢者による踏み間違え事故が社会問題となる中、高齢運転者による事故防止のためには、自動ブレーキなど先進安全技術の性能向上と普及促進が重要となる。この状況を受け、車両メーカからも、衝突軽減システム搭載車両を拡大する要求が強くなっている。その結果、より多くの車両をより短い開発期間で開発することが求められてきている。

主体は車両展開ソフトウェア開発組織であり、発表者は主体の開発リーダを担う。開発車両の増大の速さに対して現状の体制では対応が追い付かないリスクに危機感を覚え、活動を開始した。

2.改善したいこと

衝突軽減システムのソフトウェア開発では、多くの車両を開発するべく、プロダクトライン開発に取り組んでいる。組織体制としては、2 つのチーム体制を採用している。このうち「頭出しチーム」は性能向上を担っており、コア資産の作成および頭出し車両の開発をしている。もう一方の「展開チーム」は普及促進を担っており、コア資産を活用し多くの車両の開発を進めている (図 1)。発表者は「展開チーム」に所属している。

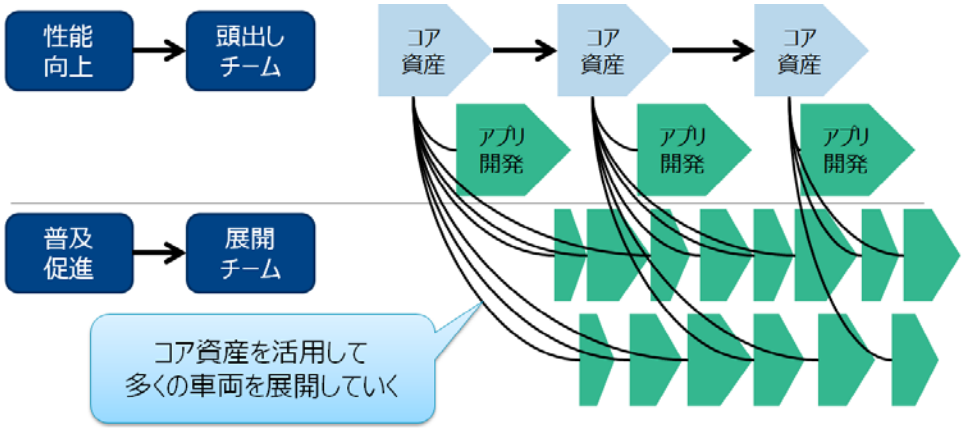


図 1.

しかし、開発車両の拡大に伴い、2016 年末、納入時期が近接する 3 車種 10 品種の同系列の開発が発生した。開発工数を見積もった結果、保有している人的リソースでは納期達成できないリスクが生じた。プロダクトライン開発での製品開発は 1 品種ごとの開発量は小さいが、同時並行開発が発生すると瞬間的な必要開発工数が大きくなる。しかし簡単には人的リソースを増やすことはできず、また開発プロセスを省くと品質を損なう恐れがある。このような制約の中、短期間で 3 車種 10 品種を開発する方法を見つけ出すことが課題となった。

3.改善策を導き出した経緯

まず我々は、この 3 車種のある共通点に着目した。その共通点とは、3 車種とも同一のコア資産から展開できる同系列のソフトウェアであるということである。同系列のソフトウェアであれば、プロセスも再利用可能である。同時並行開発の場合、再利用すべきプロセスも同時期に実行する可能性が高い。それならば、プロセスを再利用するのではなく、同時に実行することで共通部分をまとめて実行工数を省くことができるのではないだろうか、と考えた。例えば 3 車種において同一の可変点を設計するとする。この場合別々にプロセスを実行すると、毎回同じ個所の設計を検討する必要が生じる。その度にレビューなどの活動も必要となり、1 回の実行の 3 倍の工数を要する。同時に 3 車種分をまとめて実行した場合、設計で考慮すべきスコープは広がるが、設計行為やレビューなどの活動は 1 回で済むため、1 回分の工数には収まらなくとも工数が削減される効果が期待できる。

4.改善策の内容

同系列の複数車両の開発工数を低減し開発期間を短縮するため、これらの開発を単一のプロジェクトにまとめて一本のパイプラインとしてコントロールする開発方法を考案した。本開発方法は、計画と実行の 2 つのフェーズから構成される。

【計画フェーズ】

① 開発としてまとめたいプロジェクトを、簡単に仕様分析し、各プロジェクトで実行すべきプロセスをリストアップする。

② 各プロセスに存在するプロジェクト間における共通あるいは類似作業と、個別に実施必要な作業を分類する。

③ 各プロジェクトの納期に合わせて、共通作業はまとめて実施するよう優先順位を定めて計画する。

④ 個別作業については、各プロジェクトの納期を優先して最適な優先順位に配置して計画する。

【実行フェーズ】

実行フェーズでは、仕様分析、設計、実装、検査の各プロセス群に応じてプロセス実行方針を定める。

・仕様分析…発行された仕様書に対し、全プロジェクトまとめて仕様を分析する。これにより、以降の工程で重要となる「共通対応の部分」と「個別対応の部分」に識別することができる。

・設計……「共通対応の部分」の機能についてまとめて設計する。これにより、設計時間そのものを短縮できる。また「個別対応の部分」についても、仕様としては異なっているが、対応箇所としては同じ部分であれば、まとめる対象とする。

・実装……全プロジェクト分まとめて実行する。

・検査……「共通対応の部分」は先行して実行する。後に続くプロジェクトでは試験を省略し、「個別対応の部分」のみ検査を実行する。

5.改善策の実現方法

まず、今回の改善策を適用するプロジェクトについて説明する。今回のプロジェクトで開発する 3 車種は、どの車種も開発規模は同程度であり、3 か月以内に 3 車種 10 品種リリースする計画となっている。開発メンバーは 5 名である。開発フレームワークはスクラムのフレームワークを採用している。ただし、ストーリは開発要件毎ではなく、要求分析や検査などのプロセス毎に分割してバックログ化している。

このような条件下で、以下のような方策を実施した。

・従来であれば開発リーダーは 1 車種 1 名だが、3 車種まとめて 1 名の開発リーダーを割り当てた。3 車種まとめることで、全車種の情報が集約され、コミュニケーションエラーの発生を回避し、開発効率の向上が見込める。

・3 車種の仕様書が揃うように発行時期を顧客と調整した。3 車種まとめて仕様分析することのメリットを最大限に発揮することができる。

・車種毎の通信仕様対応では、同一の可変点に対する分析が繰り返される。そのため 1 車種目の分析結果を 2 車種目以降にも適用できるよう、分析結果を記録して再利用した。

・車種毎のコンフィグレーションに対しては、ソフトウェアの分析箇所は同じなので、仕様は異なっていても 3 車種まとめて分析する方針とした。

・下流工程では、「共通対応の部分」を先行して実行する。また後戻りのリスクを考慮し、新規開発アイテムを優先して試験を実施する方針とした。

・A 車、B 車、C 車のそれぞれの活動結果を取り違えないようにするため、成果物に起票する順番が常に同じになるようにガイドラインを定めた。

6.改善による変化や効果

納入日に対して 3 車種の開発活動の「共通部分」と「個別部分」を最適化することで、A・B・C 車の 3 車種の納期を達成した。また、単独で実施した D 車の工数と比較して、工数を 42%削減することができた(図 2：比較できるよう D 車の工数を 3 倍している)。

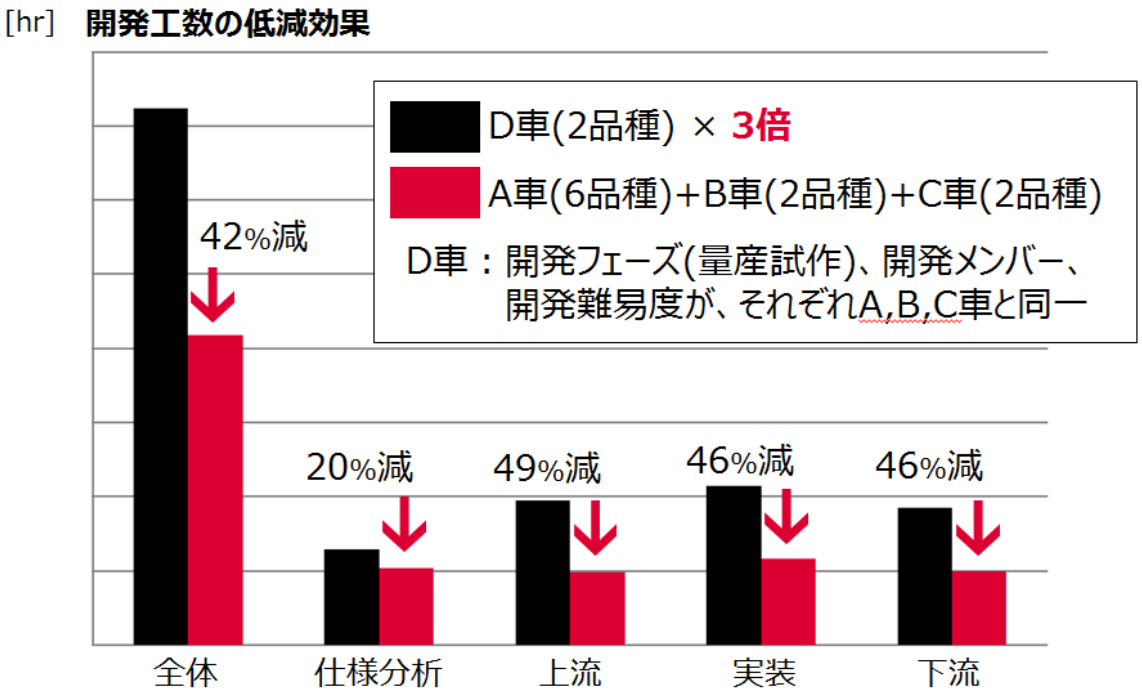


図 2.

仕様分析での工数低減(▲20%)は効果が少なかったが、これは「共通部分」の抽出および検討の工数が必要となったためである。また 3 車種同時に仕様分析をしたことから、仕様の取り違いが発生しやすい状況であり、通常よりも慎重に仕様の整理・理解を進める必要があった。上流以降の工程では「共通部分」の工数をまとめることができたため▲50%近くの工数を削減することができた。

しかし、プロセスの共通化により、開発期間は短くできたものの、その分開発リーダーの負荷が増大してしまった。開発リーダーが「成果物に対するレビュー漬け」および「 3 車種分のプロジェクトマネジメントの繁忙」という状況に陥り、次第に成果物のレビューが遅延した。そのため「成果物に対するレビューを中心に行うテクニカルリーダー」と「プロジェクトマネジメントに集中する開発リーダー」の 2 名体制に補強する必要が生じた。

7.改善活動の妥当性確認

車種展開の拡大において、今回のように開発時期が近接する事象は発生しうる。開発仕様が出揃うことが成立条件として必要であるが、このような事象に対する 1 つの方策として有効であると言える。ただし、ソフトウェアアーキテクチャを見直すことで、1 車種 1 車種の開発コストを低減することが上策である。そのような改善を果たすまでの過程において選択できる開発戦術であろう。

なお、仕様書発行について発行の遅れが生じ、それに伴う計画や作業の見直しコストが発生した。開発方針を顧客と共有し、協調して開発を進めることが必要不可欠だと感じた。また、開発リーダーの負荷について、あらかじめ対策を計画することができていなかったことで、開発現場に混乱が生じてしまった。

これらの課題を解決して、次回以降の本開発方法に取り組みたい。

A. 参考情報

[1] K. Pohl, et al., Software Product Line Engineering: Foundations, Principles and Techniques, Springer-Verlag, 2005.

[2] 林健吾, プロダクトライン開発におけるプロセスのコア資産フィードバックモデルの提案, SPI Japan 2016, JASPIC, 2016.

2D3「自動車ボディ系システムにおけるプロダクトライン開発の導入と実践」浅野雅樹（アイシン精機）

<タイトル>

自動車ボディ系システムにおけるプロダクトライン開発の導入と実践

<サブタイトル>

<発表者>

氏名（ふりがな）： 浅野 雅樹（あさの まさき）
所属： アイシン精機株式会社

<共同執筆者>

氏名（ふりがな）： 西浦 洋一（にしうら よういち）
所属： アイシン精機株式会社

氏名（ふりがな）： 中西 恒夫（なかにし つねお）
所属： 福岡大学

<主張したい点>

プロダクトライン開発の導入に成功した。2017 年 5 月末で 2 車種分の製品をすでにリリース，3～4 車種目の製品開発を実施中であり，3 製品目で黒字転換を達成できる，すなわちドメインエンジニアリングに要した費用を上回る開発コストを削減できる見込みである。

ドメインエンジニアリングの成果となるフィーチャモデルならびに構造化分析モデルによる全体理解によって，変更影響の解析に伴うコストが抑えられ，かつ開発者間のステークホルダ間の意思疎通が容易になる。アプリケーションエンジニアリングを自動化しなくとも大幅な開発コスト削減が達成できた。

イベント（Event），条件（Condition），アクション（Action）の三観点で製品群の振舞いを整理，決定図で表現し，それを共通部／可変部に分けてさらに抽象化することでリバースのフィーチャモデリングを合理的に実施できた。

<キーワード>

ソフトウェアプロダクトライン，自動車ボディ系システム，フィーチャモデル，構造化，ECA（Event-Condition-Action）

<想定する聴衆>

派生ソフトウェアの継続的な開発で既存ソフトウェア資産の構造が複雑化し，その現状理解に大きなリソースを割かれる状態にあり，その解決策としてソフトウェアプロダクトラインの導入を検討しているソフトウェア技術者。

＜活動時期＞

2014 年から現在（2017 年）に至るまで。

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（ ）

<発表内容>

1.背景

アイシン精機株式会社は，1965 年に設立されたトヨタグループに属する自動車部品メーカであるが，現在はトヨタグループ内外，国内のみならず海外の多数の一般自動車メーカ，さらには業務用輸送機器メーカに自動車部品を納めている。近年、特にボディ系製品の分野で、欧州や新興国向けの開発業務が急増しており、車両メーカ毎の個別要求機能だけでなく、仕向け地域毎のニーズや法規に適合させる必要がある為、製品毎のバリエーションが従来以上に増大する見込みであった。

これらの取り巻く環境の変化に適応するには，法や文化の異なる世界各地で，また開発思想の異なるさまざまなメーカの多様な車種で必要とされる製品を供給すべく，製品の多品種展開力を高めること，そして世界各国の開発拠点との連携できる開発体制を整えることが肝要である。図 1 はある車載製品の ECU の車種展開数，必要リソース，成り行きリソースの経緯と予測を示しているが，車種展開数の伸びに伴って必要リソースが伸びる一方であり，にも関わらず成り行きリソースは抑えられたままとなっている。

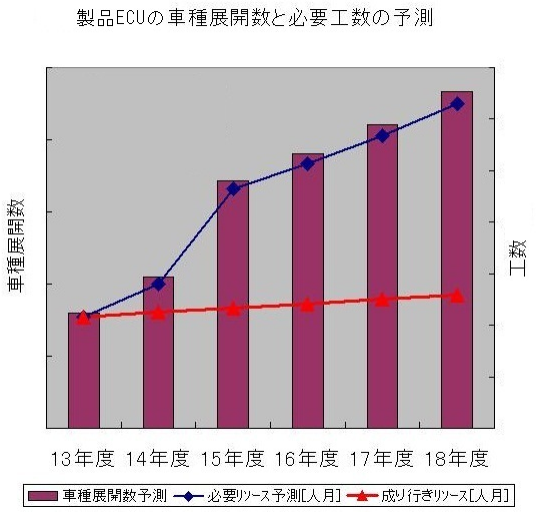


図 1

著者らの部署は，さまざまな車載ボディ系システムの制御を司るソフトウェアを開発しており，システム担当者が作成するシステム作動要求を実現する制御仕様書を作成し，それに基づいてソフトウェアを実装している。スクラッチからソフトウェアを開発することはなく，過去に開発したよく似た製品を変更する，いわゆるclone-and-ownによって派生製品のソフトウェアを開発してきた。しかしながら，過去に開発した製品のコードは特定の車種向けに依存したつくりとなっていることがしばしばあり，そのため派生製品の開発を重ねるうちにソフトウェアの構造や振舞いが複雑化し，相当のリソースをつぎ込んで影響分析をしなければ派生製品を開発できない状況になりつつあった。車種展開数は伸びる一方であり，この現状を改善しないことには競争力を維持できないとの認識を抱くようになった。

2.改善したいこと

プロセス改善活動の目標は言うまでもなく（派生製品の）開発に係る品質，開発コスト，開発期間の改善であるが，その障害となっているのは，明らかに製品間のちがいも絡んだソフトウェアの現状の構造や振舞いの理解し難さ，それに伴うシステム担当者とソフトウェア担当者の間のソフトウェアに対する現状認識の齟齬であった。システム担当者とソフトウェア開発者が変更の影響範囲に対する認識を共有するのに相当の時間を費やしていた。システム担当者，ソフトウェア担当者，またはその双方が簡単に思っている変更が，ソフトウェア実装段階において変更の影響を調べてみると予想外に大きく，実装に手間取ることもしばしば起こっていた。システム担当者，ソフトウェア担当者の双方が構造化した開発文書を共有し，ソフトウェア資産の現状を理解し，派生製品の開発を円滑にする体制やプロセスを作りたかった。

3.改善策を導き出した経緯

既存ソフトウェア資産の構造や振舞いの複雑さ，より具体的には製品のちがい，実行時の条件判断，実行時の呼出関係が大局的に整理されていないがゆえの見通しの悪さが問題の根源である。手持ちのコード資産をきれいにするだけでは不十分であり，それ以前の仕様からコードまでを一貫したかたちできれいにしておかなければ，大きな状況改善は期待できないとの認識を持っていた。

仕様からコードまでを追跡可能なかたちで整理する手段として，プロダクトライン開発[1,2]において製品間のちがいや製品間のちがいの依存関係を視覚化するためにデファクトスタンダード的に用いられているフィーチャモデリング[3,4]が有効であるように思われ，その自然な帰結としてプロダクトライン開発を導入する方向となった。

プロダクトライン開発は導入障壁，特に製品群全体で共有されるコア資産を構築するドメインエンジニアリングに要するコストが問題となりがちである。さらに既存製品群に対してプロダクトライン開発を導入する場合は，既存ソフトウェア資産をコア資産化し，アーキテクチャの現状を理解しあるべき姿を定めるべくリバースエンジニアリング（リバースモデリング）を行う必要が生じる。しかし，今回の取組みでは，何よりも既存ソフトウェアの現状を把握できるようにしたいという要求があったこと，上層部も同じ認識を持っており活動への理解があったこと，選択したボディ系システムが比較的小規模なものであったことから，プロダクトライン開発の導入への躊躇はあまりなかった。

4.改善策の内容

最初にソフトウェアプロダクトラインのパラダイム，具体的には共通性と可変性の分離，ドメインエンジニアリングとアプリケーションエンジニアリングの分離（二層開発体制），アーキテクチャ中心開発，問題空間と解決空間の分離，可変性からのトレーサビリティ確保，変化点の集中管理といった内容[2]を学習し，また社内セミナを通して社内の技術者へのパラダイムの浸透を図った。あわせて国内外企業におけるプロダクトライン開発の導入と実践の事例[5−10]を調査した。

プロダクトライン開発の導入成功が期待できそうな，すなわち極端に規模が大きくなく，それでも本稿 1 章で述べた問題を抱えているあるボディ系システム製品群を第一の導入・実践対象として選択した。

当該製品群のソフトウェアの製品間のちがいを理解すべく，制御仕様書からリバースでのフィーチャモデリングを実施するのと並行して，フィーチャモデルからの構造化分析，ならびにそうして得られた制御データフロー図やステートマシン図の要素（プロセス，制御／データフロー，イベント，アクションなど）へのフィーチャからの紐づけを試みた。そのうえで，プロダクトラインの製品群で共有するアーキテクチャをデータフロー図とステートマシン図で定義，文書化したうえで，フィーチャモデルつきのプロダクトライン版制御仕様書を作成した。

何よりもソフトウェアの理解容易化が大目標であり，コア資産からの製品導出自動化は目的ではなかったため，現状ではアプリケーションエンジニアリングについては自動化，あるいは半自動化されたプロセスが定義されているわけではない。アプリケーションエンジニアリングはプロダクトライン版制御仕様書をリファレンスとして，都度，個別製品の制御仕様書を作成することで行っている。

5.改善策の実現方法

リバースのフィーチャモデリングと構造化分析は同時並行的に行った。すなわち，リバースのフィーチャモデリングを実施してから構造化分析を行う，あるいはその逆を何度も繰り返し実施した。関係者が集まってワークショップを何度か実施していたが，おおよそのつくりはわかっていても，どのようなモデルを用いてそれをわかりやすく記述するべきか最初は定まっておらず，議論の過程でフィーチャモデルとあわせて構造化分析の制御／データフロー図，状態遷移図を描き起こす方針が定まった。

リバースのフィーチャモデリングと構造化分析を行っている過程で，ちがいの本質がシステムに入力されるボタン操作等のイベント（Event），イベントが入力されるときの条件（Condition），イベントに対するアクション（Action）の観点で整理できることが見出された。そこで根から葉にかけて，イベント，当該イベントが入力されるときの異なる条件，イベントと条件とで定まる製品によって異なり得るアクションを並べた決定図を構築し，それらの共通項の括りだしを進めて共通部と可変部を整理したうえで抽象化しフィーチャモデルを完成させた。一方，イベント，条件，アクションの整理を通じて，制御／データフロー図の整理，より

具体的には可変部の変化点としての局所化と変化点のフィーチャからの紐付けが進み，最終的に製品群を包括的に実現するアーキテクチャが定義できた。

図 2 にイベント，条件，アクション観点で製品群の振舞いをまとめた決定図の例図を示す。例は説明のために用意した仮想の自動車ボディ系システムのものである。この例では，ユーザの作動指示を受けたときの振舞いを示すが，製品モデル X では、より細かな振舞いの違いがアクションの違いに現れていることがわかる。これにより，イベントは製品で共通，イベントごとに検査される条件も製品で共通であるが，アクションが一部の製品（製品モデル X）で異なっていることが特定できる。今回の自動車ボディ系システムの実施例は，この例よりもさらに複雑なものであり，アクションのみならずイベント，条件も製品によってあったりなかったりしているが，ECA 決定木を描くことでそれらの共通部分，相違部分が整理され，また相違部分についても相違点の分解と粒度の均一化がなされていった。

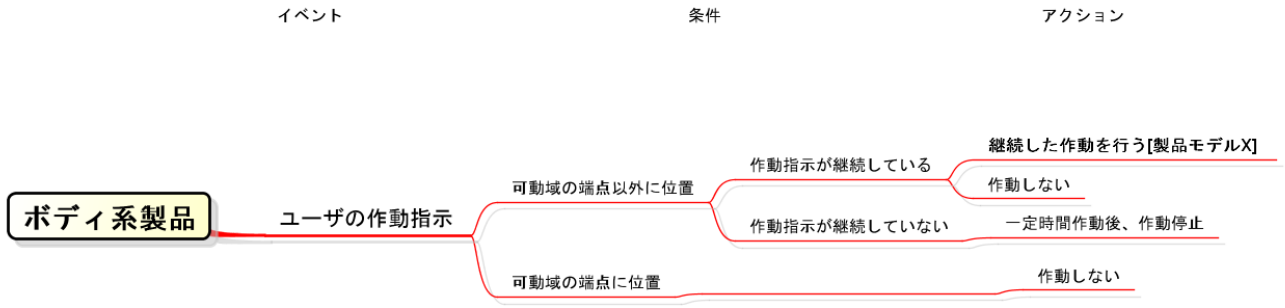


図 2: ECA 決定木例図

6.改善による変化や効果

定量的な効果として，具体的な数値は公開できないが，3 製品目にしてドメインエンジニアリングに要したコストが個別製品開発の累積コストを下回る見積もりが得られている。当該製品群は，2017 年 5 月末時点で 2 車種分の製品をすでにリリースしており，3～4 車種目の製品開発を現在実施中である。

定性的な効果として，システム担当者とソフトウェア担当者の意思疎通，特にソフトウェアに関する認識の共有が容易化されたことが挙げられる。共通部と可変部を整理した開発文書をシステム担当者に見せることにより，フィーチャ概念を介したシステムの相違点に関する議論が容易になった。それまではシステム担当者の関心を持つ変更部分に限ったコミュニケーションがなされがちだったのに対して，フィーチャモデルを共有することで，システム全体を俯瞰した上での変更点の議論ができるようになった。ソフトウェア開発者からシステム担当者への手戻り，すなわち変更点に関する見落としや誤認の後工程での発覚は体感上も実績上も減っている。

これまでソフトウェアの現状が十分に記述されておらず，開発の外部委託がやりにくい状態になっていたため，当該製品の開発はすべて本社の開発部隊で抱え込んでいたが，フィーチャモデルやプロダクトライン版制御／データフロー図，プロダクトライン版制御仕様書の完成により海外の開発部隊へすら開発業務を委託できる体制ができつつある。

7.改善活動の妥当性確認

3 製品目にして，派生製品の大幅なコスト削減が達成できそうであることから，当該製品群に対するプロダクトライン開発の導入と実践は確実な成功軌道に乗っているものと認識している。アプリケーションエンジニアリングのプロセスはいまだ「手動」であるが，その全体的あるいは部分的な自動化によって一層のコスト削減が期待できるものと考えている。プロダクトライン開発の継続性，すなわち今後，さらに派生製品を重ねていったときに，現行のコア資産やアーキテクチャのまま品質を保ち，開発コスト，開発工期を抑えた開発が継続できるかは現時点では判断できない。

本活動が他のボディ系システムにも適用できるほどの一般性があるかどうかも検討が必要である。イベント，条件，アクション観点で製品間のちがいを整理する考え方自体は一定の一般性が期待できそうだが，リバースのフィーチャモデリングと構造化分析を

同時並行的に実施する方法論についてはアプリケーションの性質や既存ソフトウェア資産の現状に依存する部分も大きく検討の余地がある。特に，異常処理の整理については，規模と複雑さを抑えつつ「きれいに」抽象化することが難しく一層の研究が望まれる。

A. 参考情報

[1] P. Clements and L. Northrop, *Software Product Lines: Practice and Patterns*, Addison-Wesley, 2001. (邦訳: 前田 卓雄（翻訳）,『ソフトウェアプロダクトライン: ユビキタスネットワーク時代のソフトウェアビジネス戦略と実践』, 日刊工業新聞社, 2003.)

[2] K. Pohl, G. Böckle, and F. v. d. Linden, Software Product Line Engineering: Foundation, Principles, and Techniques, Springer, 2005. （邦訳: 林 好一, 吉村 健太郎, 今関 剛（訳）,『ソフトウェアプロダクトラインエンジニアリング: ソフトウェア製品系列開発の基礎と概念から技法まで』, SiB アクセス, 2009.）

[3] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study," Technical Report CMU/SEI-90-TR-21, SEI/CMU, Nov. 1990.

[4] K. C. Kang, S. Kim, J. Lee, K. Kim, G. J. Kim, and E. Shin, "FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architecture," Annals of Software Engineering, Vol.5, pp.143–168, 1998.

[5] F. v. d. Linden, K. Schmid, and E. Rommes, *Software Product Lines in Action*, Springer, 2007.

[6] M. Steger, C. Tischer, B. Boss, A. Müller, O. Pertler, W. Stotz, and S. Ferber, "Introducing PLA at Bosch Gasoline Systems: Experiences and Practices," Proc. Software Prodcut Line Conf. 2004, pp. 34–50, Aug. 2004.

[7] Y. Takebe, N. Fukaya, M. Chikahisa, T. Hanawa, and O. Shirai, "Experiences with Software Product Line Engineering in Product Development Oriented Organization," Proc. Software Product Line Conf. 2009, pp. 275–283, Sep. 2009.

[8] T. Iwasaki, M. Uchiba, J. Ohtsuka, K. Hachiya, T. Nakanishi, K. Hisazumi, and A. Fukuda, "An Experience Report of Introducing Product Line Engineering across the Board," Proc. 14th Int. Software Product Line Conf. (SPLC) 2010, Vol.2, pp.255–258, Sep. 2010.

[9] 岩崎 孝司, 内場 誠, 大塚 潤, 八谷 浩二, 中西 恒夫, 久住 憲嗣, 福田 晃,「プロダクトライン開発手法の全社的導入に関する一事例報告」, 組込みシステムシンポジウム 2010（ESS2010）予稿集, 2010 年 10 月.

[10] 大塚 潤, 河原畑 光一, 岩崎 孝司, 内場 誠, 中西 恒夫, 久住 憲嗣, 福田 晃,「大規模移動体ネットワーク機器ファームウェア開発へのソフトウェアプロダクトライン適用事例」, 組込みシステムシンポジウム 2011（ESS2011）予稿集, 2011 年 10 月.

3A1「急成長プロダクトの Dev&Ops で生じる悪循環とその解決策」横山翔 (リクルートテクノロジーズ)

<タイトル>

急成長プロダクトの Dev&Ops で生じる悪循環とその解決策

<サブタイトル>

<発表者>

氏名（ふりがな）：横山翔
所属： リクルートテクノロジーズ

<共同執筆者>

氏名（ふりがな）：
所属：

<主張したい点>

問題

急成長フェーズのプロダクトにて、開発・運用チームが疲弊し「不透明性→局所最適な行動→品質劣化→相次ぐトラブル→さらなる疲弊」の悪循環に陥った。

打ち手

検査と適応を繰り返した。

- サービスレベル定義→品質・障害対応の目線合わせ
- スプリントセレモニー開催→会話の活性化
- バリューストリームマップ作成→業務の可視化
- ドキュメントのデザインパターン化→知識共有の促進

結果

「可視化→全体最適志向→品質予防＋SLA・OLA に基づく対応→安定化」の好循環が回り始めた。

(248 文字)

<キーワード>

プロダクト、グロース、スケール、再構築、局所最適、ボトルネック、ムダ・ムラ・ムリ、形骸化、サービスレベル、オペレーションレベル、スクラム、KPT、1on1、バリューストリームマップ、ドキュメント、推進、可視化、共通認識、改善

<想定する聴衆>

急成長フェーズのプロダクト開発・運用に関心のあるソフトウェアエンジニア、チームリーダー、組織マネージャー。

＜活動時期＞

2016 年 10 月～継続中。主に 2017 年 4 月までの成果を共有します。

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動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

企業

上位組織の目的

会社の次を担うプロダクトとして成功させたいと考えている。

- プロダクト：年率 300%で売上が伸びている
- チーム：4 半期ごとにメンバー数を倍増する

前提、動向：

- ヒットするか不確定だった初期フェーズでは、外注への丸投げでシステムは低品質だった
- 急成長に伴って限界が来たためサブシステムを順番に再構築することにした（Dev）
- 同時並行で既存システムの保守運用を行っていた（Ops）
- 置き換えが完了したサブシステムは順次、保守運用の担当に受け渡されて、機能追加を再開した（Dev&Ops）

発表者の役割

- 保守運用チームを立て直すこと
- その上でさらなるグロースに向けた土台を構築すること

きっかけ

- トップダウンによる指示で実情を調査し、改善の必要性と推進役を担うことを訴えた

2.改善したいこと

「局所最適な行動→品質劣化→相次ぐ深夜休日のトラブル対応→チームの疲弊」の悪循環が生じていた。

トラブル対応・疲弊

- システム障害が 1 営業日あたり 2 件発生するという状態が 1 ヶ月続いた。
- 不眠での深夜対応や休日出社が常態化していた。

局所最適化・品質劣化

- 担当者以外が各自の作業内容を把握していない状態だった。
- そのためレビュー、相互サポート、不在時の緊急対応ができておらず、品質が劣化しやすい状態だった。
- また、チームのスケールに向けた継承の準備ができておらず、さらなる品質悪化が懸念された。

3.改善策を導き出した経緯

現状の可視化→ボトルネックの特定→プラクティスの適用を繰り返した。

＃ 障害の傾向分析から着手した

- 不適切な対応による二次災害が少なからず発生していた
- 軽微な不具合でのオンコールや残業対応が発生していた

→ サービスレベルを整備！

＃ 次に 1on1 でのヒアリングを実施した

- 継続的な課題検知・課題解消の座組が必要だった
    - メンバーごとに異なる課題意識を持っていた
    - 互いに問題を発信しあえていなかった
    - チームとしての問題解決に着手できていなかった
- リソース固定のもとで優先順位の高いものから対応する座組が必要だった
    - 「全ての作業が高い優先度」として計画が立てられていた・割り込み作業が発生していた
    - 稼働を増やすことで計画に実態を合わせようとしていた

→ スクラムのプラクティスを現場が受け入れられる順番で徐々に適用！

＃ KPT を通して課題が浮上した

＃＃ 全体像を全く知らない

- あらゆる開発・運用活動の、あらゆる工程で、あらゆるトピックの課題が浮上した
- 全体像が不透明だったので、優先順位決め（＝ボトルネック特定）や相互サポートが困難だった
    - 担当者以外は（時には担当者自身でさえ）各々の業務フローを把握できていなかった
    - それぞれの業務内容のどこに、どのようなムダ・ムラ・ムリが生じているのか見えなかった

→ 各業務のバリューストリームマップを作成！

全体像にアクセスできない

- セレモニーでの改善活動を通してルール・プロセスが決まっていくが、意思決定スピードにドキュメント化が間に合わない

→ ボーイスカウト原則の普及！（通り掛かったメンバーがドキュメント化する・綺麗にする）

- 各メンバーが自発的に作業手順を文書化するようになり、既存のプロセスを改善しようとした結果、別の場所に似たようなドキュメントが二重・三重にできてしまう
- A さんが作業ミスを反省してリリース手順書をアップデートしても、B さんは別のリリース手順書を見ているので同じミスをしてしまう
- プロセスを急激に整備する中で、メンバーが混乱したとき・迷ったときにどこを見れば良いか分からない

→ ドキュメント構成を（ソースコードと同じように）デザインパターンとして設計・リファクタリング！

4.改善策の内容

サービスレベルを定義した

品質・障害対応に対する関係者の目線を合わせた。
定義そのものよりも定期的な会話を通した共通認識の醸成を重視した。

- SLA：機能（コア機能・サブ機能）、非機能（可用性・性能）
- OLA：SLA に応じた対応速度、影響範囲に応じた対応速度（利用者数・時間帯・サポート環境）
- 運用：定期的な評価、再発防止、予防、定義自体の見直し

スプリントセレモニーを開催した

チーム内部での会話を活性化させて透明性を保つ。
その上で、検査・適応のサイクルを回せるようにした。

- レトロスペクティブ：プロセスの再計画
- デイリーミーティング：作業の再計画
- 週次プランニング：案件の再計画
- 週次レビュー：作業結果のフィードバック

また、セレモニー開催に伴ってアイテムを整備した。

- プロダクトバックログ：限られた人員で優先順位に従って作業を実施するためのキュー
- スプリントバックログ：現状の作業進捗を可視化するカンバン
- 障害物リスト：レトロスペクティブで可視化された課題の一覧

＃ バリューストリームマップを作成した

各担当者にヒアリングしながら全員で業務フローを可視化した。
作業時間・待ち時間や手戻りの実態を洗い出し、ムダ・ムラ・ムリが生じる箇所を特定した。

- 機能追加開発
- システム障害対応
- ユーザーからの問い合わせ対応
- 連携部署からの調査依頼対応
- 連携部署からのサービス運用作業依頼対応
- EOSL 対応などのシステム保守業務
- キャパシティ計測などのシステム運用業務

＃ ドキュメントをソースコードのように扱う

どこにどのようなドキュメントがあるべきかを、コーディングに例えて整理した。
当たり前と思える文書管理 Tips であっても、チーム全体で認識を揃えることで、知識共有を促進した。

- コードレビュー・ペアプログラミング：ドキュメント構成に悩んだときはレビュー依頼やディスカッションする。
- ボーイスカウト原則：気付いた人が気付いたときに加筆・修正する。文書を見る前より、見た後の方が綺麗な状態を目指す。
- リファクタリング：初期設計自体に問題があったら、継続的に修正していく。
- MVC アーキテクチャ：（例）集計結果を関係者に見せるドキュメント→元となる記録のドキュメント、といった依存構造で分離する。
- デザインパターン：（例）リリース作業記録は Template から必ずコピーして、親ディレクトリの下に Iterator として追加していく。

5.改善策の実現方法

＃ サービスレベルの定義に当たって

- 記述方法・標準化
    - 文献や社内事例をもとにドキュメントを整備した
    - アップデートしやすいようにコラボレーションツール（Confluence）上に記述した
- これを決めて何の意味があるの？という声
    - 過去のシステム運用実績・トラブル事例を踏まえ「このトラブルだとこういう対応になります」をシミュレーションした
    - 導入時に認識を合わせた
- サービスレベルではなく感情や焦りで意思決定がなされがち
    - トラブル発生時には開発・運用メンバーから発表者に一声掛けるようにした
    - 声を掛けやすくなるように、メンバーが一方的に不利な立場にならないようなコミュニケーションを重ねた

    - 意思決定時に「サービスレベルだとこのように判断できますね」という会話を重ねた
    - 判断の寄る辺として定着させた
- 初期には考慮されていない新観点が出るとサービスレベルが機能しなくなる
    - 月次の振り返りで定義自体を加筆・修正した
    - 過大・過少なオペレーションレベルで対応したのであれば、次からは適切なレベルとする旨を会話した
    - 形骸化・硬直化を防ぐような運用設計にしている

スプリントセレモニーの開催に当たって

- レトロスペクティブ（KPT）から始めた
    - 厳しい状況下だからこそ発揮された良い点をしっかりと称え合った
    - その上で各人が感じている課題を洗い出した
- Try としてプラクティスを導入した
    - Problem：1 度にあれもこれも対応するのは無理だ！
        - 優先順位をキューで管理するためにバックログを導入
        - キューの見直し機会としてプランニング会議を導入
    - Problem：他の人が何をやっているのか分からないからサポートできない！
        - デイリーミーティングや週次レビューを導入
- プラクティスの適用が急すぎて、メンバーが咀嚼できなくなった
    - 納得できる範囲を聞き出して、全員が納得できる部分から徐々に導入した
    - 中途半端な適用ゆえの Problem が挙がるので、そのタイミングで Try としてさらに適用した
    - 「根っこの考え方を知りたい」という意見が挙がったタイミングで、希望者に対して参考図書の案内や感想共有を実施した

バリューストリームマップの作成に当たって

- ワークショップ形式で実施した
    - 問題意識・打ち手を端的に説明＋社内事例（ワークショップ中の写真などを交えて）を紹介して納得してもらう
    - 全員で作成した→ムダ・ムラ・ムリや認識差分が可視化された→まさに進行中だった案件の問題を検知できた→業務改善の成功体験
- あらゆる業務フローを可視化するには拘束時間が多すぎるとの意見が出た
    - 個別に担当者ヒアリングを実施して各種業務のバリューストリームマップを作成した
    - 明らかにすぐできる業務改善のみ実施して、本格的な改善はチームが本格的にクロスファンクショナル化したあとに申し送り

ドキュメントをソースコードのように扱う

- ボーイスカウト・コードレビュー・リファクタリング
    - Problem として挙がった意見を踏まえて、ドキュメント化＋更新し続けることの重要性を伝えた
    - セレモニーや普段のコミュニケーションの中で「その知見をぜひ文書化してください」「更新してください」とひらすら言い続けた
    - どうやったら促進できるかメンバーに案を募った→bot やチケット運用など試行錯誤してもらった→チーム全員に推進

者として振舞ってもらった
- アーキテクチャ・デザインパターン
    - Problem として挙がった意見を踏まえて、ソースコードと同じように綺麗に設計しようと伝えた
    - 「こういう意図でここに書いたけど設計として良さそうですか」といった相談をする Slack チャンネルを設けてレビューを活性化させた
    - チーム混乱の最大のボトルネックがルール不備≒ドキュメント構成となってからは、発表者は文書の再構成・分離結合に専念した

6.改善による変化や効果

「局所最適な行動→品質劣化→相次ぐ深夜休日のトラブル対応→チームの疲弊」の悪循環が抑えられた。
「可視化→全体最適志向→品質予防＋SLA・OLA に基づく対応→安定化」の好循環が回り始めた。

トラブル対応・疲弊 → SLA・OLA に基づく対応・安定化

- 重大なシステム障害は 1 ヶ月当たり 0.5 件にまで減少した
- 軽微なシステム障害に対しては、不眠での深夜対応や休日出社を行わないようになった

局所最適化・品質劣化 → 可視化・全体最適志向・品質予防

- 担当者以外が各自の作業に助言やサポートを行うようになり、事前に問題点を検知できるようになった
- 可視化を通して、チームのスケールに向けた継承の準備にもなり、新規参画者からは「ドキュメントやサポートが充実していた。こんなにスムーズに立ち上がれたのは初めてだ。」との Keep が挙がった（3 ヶ月前の新規参画者は正反対の Problem を挙げていた）

7.改善活動の妥当性確認

発表者の役割

- 保守運用チームを立て直すこと
    - 安定化は実現できた
    - 共通認識を醸成するために、現場メンバーのコミュニケーションコストを大幅に費やした
    - 上記コストは必要だったと上位組織・発表者は認識しているが、ステークホルダー全体との握りは弱かった
    - 結果として上位組織や他部署メンバーに裏でフォローいただく形となった
- その上でさらなるグロースに向けた土台を構築すること
    - 業務フローの可視化やプロセスの整備という観点では土台構築に貢献できた
    - 一方で、以下の課題が残った→現在はこれらの課題に取り組み中
        - システム観点での土台構築が不十分である（例：技術的負債を予防する仕組みが設けられていない）
        - 次の四半期で倍増するメンバーが経緯を知らないためプロセスに納得感を持つことができない＆改善の提案をしにくい

- Dev/Ops の安定化に特化したため、Biz/Dev/Ops の全体最適への方針が見えておらず、改善活動が局所最適化を促進している可能性がある
- Dev/Ops のさらなる最適化（例：チームのクロスファンクショナル化）の余地が残っている

A. 参考情報

[1]『リーン開発の本質』
[2]『スクラムガイド』
[3]『組織パターン』
[4]『エクストリームプログラミング』
[5]『ウェブオペレーション』
[6]『強い会社はこうして作られる！ –ITIL 実践の鉄則』
[7] https://speakerdeck.com/bufferings/jie-guo-de-nisukuramuninatuteru-nafalsegaiitosi-u-number-rsgt2017
[8] https://docs.com/ushio-tsuyoshi/8263
[9] https://www.slideshare.net/i2key/devsumib
[10] http://yuzutas0.hatenablog.com/entry/2017/05/23/073000
[11] http://yuzutas0.hatenablog.com/entry/2017/07/06/083000

3A2「DevOps でリードタイムを８ヶ月 から最短１週間まで短縮！！」安藤寿之（NEC ソリューションイノベータ）

<タイトル>

DevOps でリードタイムを８ヶ月 から最短１週間まで短縮！！

<サブタイトル>

マネージャや開発チームの赤裸々話

<発表者>

氏名（ふりがな）：安藤寿之（あんどうとしゆき）
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）： 福井知宏（ふくいともひろ）
所属： NEC ソリューションイノベータ株式会社 情報システム部

<主張したい点>

超保守的なエンプラ情シスが DevOps を実践するまでの道程(**J**ourney)

<キーワード>

DevOps

<想定する聴衆>

「エンプラ」のややこしい環境で四苦八苦されている開発者、マネージャの方。
もしくは、新しい考え方やプロセスを導入しようと四苦八苦されているスタッフ部門の方。

＜活動時期＞

2014 年 12 月～2017 年 3 月

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（ ）

<発表内容>

1.背景

長年にわたって数多くのシステムを追加してきた結果、各システムがサイロ化し、部門間の情報共有がうまくできていない、という問題が生じるようになっていた。このような状況の中、従来のウォーターフォール型開発では、期間システムの改修に半年～１年程度を要していた。また、リリースした際に、事業部門から「ちょっと違うんだよなぁ」という反応を頂くことが多々あった。約 1 億 3,000 万円かけて開発したアプリケーションの機能のうち 約 20% がユーザー要件を満たしていないことがわかり、その後さらに 3,000 万円の追加投資をしたというケースもあった。

2.改善したいこと

以下の 2 つの観点で、情報システム部の開発チームと運用チームのモチベーションを向上する。

- 新しい開発方法と技術を通じて、自らが成長していると実感できる
- 事業部門からの頻繁なフィードバックと改善を通じて、事業部門が満足するシステムと提供していると実感できる

3.改善策を導き出した経緯

アイデアが機能となり、実際に利用できるまでの開発プロセスを、バリューストリームマップとして見える化した。

4.改善策の内容

人、プロセス、プロダクトの観点で、それぞれ以下を実施した。

- 人：DevOps、アジャイルのベースとなる「西洋文化のマインドセット」の取り込み
- プロセス：開発・運用の連携と自動化による承認、作業フローの簡略化
- プロダクト：マイクロソフトのクラウドサービス(Visual Studio Team Services、Azure)とオープンソース ソフトウェア(OSS)を組み合わせたビルド・リリースの自動化基盤の構築

5.改善策の実現方法

バリューストリームマップにより見える化したプロセスに対して、そこにある数々のムダを削減して、プロセスを改善することで得られる具体的な成果(リードタイムが 8 ヶ月から 1 週間)を幹部、マネージャ、開発者を含めて共有した。
そして、従来の慣習・プロセスを離れゼロベースで検討するために影響の小さいプロジェクトを特例プロジェクトと定め、人/プロセス/プロダクトの観点で以下の取り組みを行った。

- 人：「西洋文化のマインドセット」の取り込み
  DevOps やアジャイルのベースとなる「西洋文化のマインドセット」について、マイクロソフトのエバンジェリストが開催する講義や、ワークショップを受講し、その後のプロセス改善を進める上での参考にした。

- プロセス：開発・運用の連携と自動化による承認、作業フローの簡略化
  リリースは自動化を基本とし、手順書レス、かつ、最低限の承認のみに絞り込んだ。

- プロダクト：マイクロソフトのクラウドサービス(Visual Studio Team Services、Azure)とオープンソース ソフトウェア(OSS)を組み合わせたビルド・リリースの自動化基盤の構築
  これまで利用していたオープンソース ソフトウェア (OSS)にマイクロソフトのクラウドサービスを組み合わせ、ビルド・テスト・リリースの自動化を進めた。

6.改善による変化や効果

平均 8 か月かかっていたリリースサイクルを、最短 1 週間と大幅に短縮、事業部門の要望を取り入れた改修を短いサイクルで実現できるようになった。

- 人：これまでの習慣にとらわれず、新しい観点で考えられるようになった。
- プロセス：自動化により手順書レスになり、運用チームの承認が簡略化した。
- プロダクト：要求からリリースまでのトレーサビリティが高まった。

7.改善活動の妥当性確認

事業部門から「実物を見てようやく具体的なイメージが持てた」とのお言葉をいただくことができた。

また、情報システム部の開発チームと運用チームは、外部の勉強会などにも参加し、新しい開発方法や技術を仕事に活用することを検討するようになった。

A. 参考情報

[1] 社内システムの開発/運用に Microsoft Azure と OSS で構成した DevOps を導入。以前は平均 8 か月だったリリース サイクルを最短 1 週間にまで短縮、初期リリースまでのコストも 1/4 に ～職場環境改善ツール「Workplace Environment Improvement Tool (WEIT)」をわずか 3 か月でリリース～
https://customers.microsoft.com/en-us/story/necsolutionjapanese
※ マイクロソフト社にて公開されている本事例の記事。

3A3「基幹システムの開発・保守における機械学習の適用検討とその評価」陣内孝司（住友電工情報システム）

<タイトル>

基幹システムの開発・保守における機械学習の適用検討とその評価

<サブタイトル>

なし

<発表者>

氏名（ふりがな）：陣内　孝司(じんない　たかし)
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　第一システム部第二開発グループ

<共同執筆者>

氏名（ふりがな）：　中村　伸裕(なかむら　のぶひろ)
所属：住友電工情報システム株式会社
ＱＣＤ改善推進部

<主張したい点>

機械学習をうまく活用すれば、以下のようなメリットが得られるため積極的に活用すべき
・基幹システムにおいて、従来にはなかった価値を提供することが可能
・システム保守において、顧客満足につながるシステム改善提案の契機が得られる
・必要な文書を効率的に探せ、開発・保守の工数削減につながる
・プログラマが作り込んだ欠陥数を予想でき、プロジェクトとしての品質保証の拠り所ができる

<キーワード>

機械学習、システム改善提案、欠陥数予想、価値提供、業務効率化

<想定する聴衆>

開発者、プロジェクトマネージャ、システム保守担当、改善推進者

＜活動時期＞

・2016 年４月～2017 年５月(継続中)

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
■ 改善活動の結果が明確になっている段階
□ その他（　　）

<発表内容>

1. 背景

現在は、様々な分野で AI ブームが起きている。当部門では以下の目的で 2016 年 4 月より機械学習チームを設置し、活動を進めてきた。

・基幹システムの機能として AI の適用方法を具体化する

・システム開発、保守を支援する機能として AI の適用方法を具体化する

2. 技術習得

使用するツールは以下の理由で「Apatch Spark」を選定した

・当社製品「楽々FrameworkⅡ」が Java で構築されており連携が容易である

・分散処理をサポートしている

・IBM がサポートしているため長期的なサポートが期待できる

・人気が高い

・今後も機能拡張が期待できる

Spark の情報を書籍・WEB で収集し、実際に動かして技術を習得した。

日本語のページがあまりなく、うまく動かないときの対処に苦労したが、

3～4 カ月程度である程度自由に使用できるようになった。

3. 適用テーマの検討

適用テーマを選定するために、各部署から合計 15 名参加いただき、説明会を実施した。その後、参加者にどんなことができると嬉しいか、要望を考えてもらった。その結果、テーマが 31 件収集できた。以下に代表的な例を示す。

・PG 開発規模見積算出

・ＳＱＬ性能問題を事前検知

・レビュー指摘から今後発生する問題、対策をアドバイス

・ネーミング揺らぎ抽出

など

この中から機械学習チームが実装方法を検討し、以下のテーマに取り組むこととした。

・画面遷移評価による非効率な機能の改善提案

システムのログから画面遷移を学習し、標準的な画面遷移を求める。非効率な機能はエラーが発生したりして標準的な画面遷移から外れることが多い。このような機能を自動抽出し改善提案する。

・勤務報告システムの実績登録の効率化

作業分類コードをレコメンドし、選択にかかる時間を削減

・文書を探す時間の短縮

例えば会議中に話の流れで予定外に説明してもらった文書を、会議終了後確認したり、議事録を作成したりする際、保管場所がわからないことがある。このような文書の URL をレコメンドして業務の効率化を図る。(当組織では、大半の文書が WEB 化されており、必要に応じて会議中に閲覧できる。)

・ソースコードの Metrics から作込欠陥数予測

過去に作成されたソースコードの Metrics と欠陥数を機械学習し、今後作成するソースコードの欠陥数を予測する。

次章で取組内容を説明する。

4. 機械学習の評価

4.1 画面遷移評価による非効率な機能の改善提案

(a)背景

当組織の 2017 年の組織目標として保守提案数が設定されている。具体的な提案アプローチの一つとして、システムログから使い勝手の悪い機能を見つけて改善提案するという施策が設定された。

(b)改善策

当部門で開発したシステムは、「楽々Framework Ⅱ」のログ機能を使用し、利用ユーザー、利用日時、利用機能、画面番号等が記録されている。そのログを活用し、エラーにより画面が戻るといった非効率な機能を自動抽出するツールを開発・提供する。保守担当者はツールが出力するプログラムの改善方法を考え提案する。

(c) AI の適用方法

Spark は、GraphX というグラフ(「ノード」と「ノード間の関係」)を扱う機能を持っており、画面遷移の前後関係を表現することができる。図 1 に画面遷移グラフを示す。ノードは画面の種類としている。矢印は画面間の遷移を示しており、矢印付近の数字は画面遷移回数を表している。遷移回数の多い経路を標準画面遷移として学習する。(×印は学習の結果、捨てられた画面遷移である。)

次に、一つの業務を完了するのに標準的な画面遷移が行われたのか、エラー等により余分な画面遷移が発生したか、計算する。標準的な画面遷移が行われた割合を標準画面効率指数とし、低いほど非効率な業務となっている。

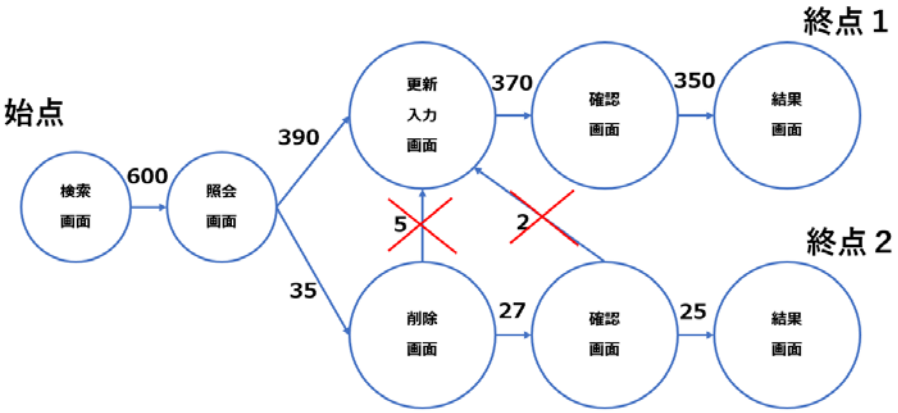


図 1：画面遷移グラフ

表 1 に、ツールの出力例を示す。各行はユーザーID に示された作業者が標準画面遷移通りに遷移しなかった機能および、業務を示している。標準画面遷移効率指数の低いものから改善方法を検討し、ユーザーに改善提案を行う。この場合は、ユーザー A が購入依頼状況照会の照会の際の検索で何らかのエラーを頻発し、10 回に 1 回の割合でしかスムーズに操作できなかったことを示している。そういった機能に対して、問題点をヒアリングすることで改善提案を行う。

表 1 ：標準画面遷移効率指数一覧

機能名称	業務内容	ユーザー	標準画面遷移効率指数
購入依頼状況照会	照会	A	0.1
原価承認	更新	B	0.2
原価承認解除	更新	B	0.2
原価計算	登録	B	0.4
設計完了登録	更新	B	0.4
能力負荷	更新	B	0.9

(d)結果

試行プロジェクトで標準画面遷移指数が低いプログラムのヒアリングを行った。
その結果、用意していた改善提案は採用されなかったが現場で不便に思っていることが把握でき、
新たに 5 件の改善提案を行っている。直接的な改善提案にはならなかったが、
ヒアリングの糸口となる意味で、価値にあるツールのなったと考えられる。

4.2 勤務報告システムへのレコメンド適用

(a)背景

当組織では、システム開発の実績時間をプロジェクト毎に外部設計やコーディング、単体テストといった作業分類別に入力する決まりとなっている。しかし、作業分類は 400 近くの分類があり、ウインドウ 1 ページに収まらず、ページを繰って探している。
勤務報告システムは、二週間以内に自身が登録した内容を記憶しておき、その作業分類は実績時間を簡単に入力できる。しかし、プロジェクトのフェーズが変わった時などは、新たに作業分類コードを選択する必要があり、毎回 5 秒～10 秒程度のタイムロスが発生している。この一人当たりロスは大した時間ではないが、登録人数が多いため組織全体では、1.2H/日程度の工数ロスが発生している。

(b)改善策

似たような作業者が最近使用した作業分類コードをレコメンドし、ポップアップに表示、選択させることで登録時間削減を狙った。

(c)AI の適用方法

機械学習の手法として ALS(協調フィルタリング)があり、ショッピングサイトの「あなたへのおすすめ」などで利用されている。図 1 は ALS の概要を示したものであり、縦軸はユーザー、横軸が製品を表す。図 1 のセルは Rating であり、通常購入回数やお気に入り度(星の数)が利用される。ALS は数値のあるセルのデータから、数値のないセルのデータを予測する手法である。例として表 1 から表 2 の値を予測する。赤字はもともと存在しなかったデータを ALS が予測した値となっている例である。

表 2． 実データ

user \ product	product1	product2	product3	product4	product5	product6	product7
user1		1.00	2.00	2.00	1.00	4.00	10.00
user2	1.00	1.00	2.00	2.00	1.00	4.00	5.00
user3	2.00				2.00		10.00

表３：予測データ

user \ product	product1	product2	product3	product4	product5	product6	product7
user1	1.01	1.01	2.02	2.02	1.01	4.01	11.00
user2	1.01	1.01	2.02	2.02	1.01	4.01	5.20
user3	2.01	2.01	4.02	4.04	2.00	7.96	10.00

今回適用する勤務報告システムでは、重みを示すデータを持っていない。
最近使用した作業分類コードが、よりレコメンドされるように以下の式で Rating を与える様に工夫した。

Rating　＝　1　/（当日　－　登録日）

また、Spark が提供する ALS の API は、user、product ともに Long 型 1 項目となっている。Product には、作業分類 ID を割り当てた。プロジェクト ID は学習していないため、同一プロジェクトを担当していることが学習できないので、レコメンドの質が低下すると考えた。そこで作業分類 ID と重複しないプロジェクトを識別する ID を使った疑似データを生成し、学習させた。

(d)結果

レコメンド結果を評価するために、評価対象者を 4 名ピックアップし、該当者が当日初めて入力した作業分類コードがレコメンドされているか確認した。その結果、3 名が入力した作業分類コードが 20 位以内(1 ページ目)に入っていた。残り 1 名は 135 位と 352 位であった。レコメンドがよく当たる人と当たらない人がいるものの、組織全体としては効果があると判断し、17 年 4 月より本番運用を開始した。なお、32 位と 70 位のデータは疑似データなしの場合、それぞれ 148 位、412 位であり、疑似データの効果を確認できた。

4.3 お探し文書のレコメンド

(a)背景

当組織では、システムドキュメントや改善活動の議事録等が約 82 万文書存在し、IS Portal と呼ばれる文書サーバーで統合管理されており、社員はブラウザですべての文書を閲覧することができる。
システム開発やワーキンググループ活動で会議中に見せてもらった文書を後で確認しようとした時に、文書を探すようなことがよくある。探している文書がレコメンドされると作業効率を上げることができる。

(b)改善策

当 Web サーバーのアクセスログを学習し、ALS の機能を使用しておすすめの文書を提案する。

(c)AI の適用方法

4.2 と同様の協調フィルタリングを使用し、ユーザーと文書 URL の関係を学習させる。
学習結果から、ユーザー毎に文書の URL をレコメンドする。

(d)結果

現在開発中であるが、試しに開発者 2 名のレコメンドを行った結果、仕事に関係の深い文書が提案されており、本番化の準備を進めている。

4.4 ソースコードの構造からの欠陥予測

(a)背景

機械学習の適用例を調査している最中に、“A Machine Learning Based Model For Software Defect Prediction”[1]を発見し、ソースコードのメトリクスから欠陥数を予測していた。図 2 は、この論文で示されている図で、実際の欠陥数と予測値の関係を示したもの(横軸が欠陥数、縦軸が予測値)である。高い的中率であったため、当組織で適用できれば開発における品質改善に役立つと判断し、試行することとした。

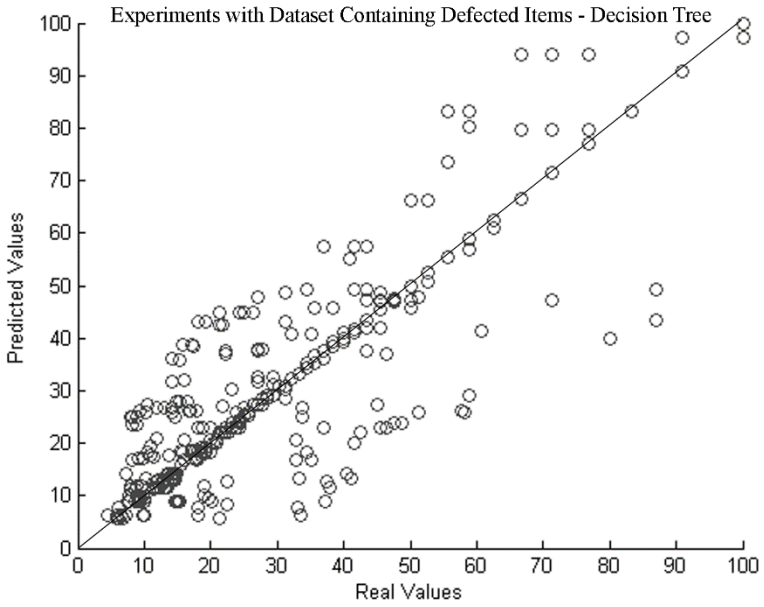


図2 : The predicted values and the real values in decision tree experiments where the input data set contains only defected items[1]

(b)改善策

プログラム完成後、ソースコードのメトリクスから欠陥数を予測し、その値を目標としてテストを実施する。当組織で利用可能なメトリクスとしては、行数・複雑度・階層の深さ等がある。

(c)AI の適用方法

参考論文では Decision Tree を使用していたが、その後、より性能の高い Random Forest が普及しているため、Random Forest を使用することとした。メトリクスは、各プロジェクトに依頼し、Eclipse のプラグインである「Metrics プラグイン」から取得してもらった。正解データとしては、毎年集計している品質ベースラインの明細データを利用した。

(d)結果

約 500 本のプログラムを学習させ、予測させた。その結果、論文の様な予測精度にはならず、本番化を断念した。

当初予想では、複雑なプログラムほど高い欠陥数を記録すると予想していた。しかし実際は、複雑になるほど欠陥数が安定する結果が出た。通常簡単なプログラムを初心者に割り当て、難しいプログラムを上級者に割り当てるため、このような結果となったと考えられる。

今後 Deep Learning の使用や学習データの追加などを行い、再挑戦する予定である。

５．まとめ

機械学習は、技術的難易度が高いと思っていたが、実際にやってみると意外と簡単で、効果が得られることが分かった。
しかし、結果はデータ次第であることを改めて実感することとなった。
今後、システム開発・保守の分野で Deep Learning や分散処理の技術を応用し、
事業部の業績向上や、開発プロセス改善に貢献するよう、様々な適用方法を考えてチャレンジしていきたい。

A. 参考情報

[1] Onur Kutlubay, Mehmet Balman, Doğu Gül, Ayşe B. Bener Boğaziçi University, Computer Engineering Department , “A Machine Learning Based Model For Software Defect Prediction”

3B1「自分事化影響要因に着目した中期経営計画立案・展開への共創アプローチ[現状分析～計画立案編]」安達賢二(HBA)

<タイトル>

自分事化影響要因に着目した中期経営計画立案・展開への共創アプローチ[現状分析～計画立案編]

<サブタイトル>

<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>

氏名（ふりがな）：
所属：

<主張したい点>

弊社の創立 50 周年に掲げた新テーマ"IT で幸せに挑む"の実践・実現に向けて、中期経営計画の立案を進めるうえで、これまで行ってきた経営企画部門と役員クラスの事前調整に基づくトップダウンアプローチではなく、"モノゴトの自分事化に影響を与える要因"に着目した全社員参画型の共創アプローチを採用した。
役員～実務担当者までの多くの社員を巻き込み未来の会社像と現状のギャップを埋める施策検討を行った結果、社員の想いとトップの想いを融合した中期経営計画を作成できた。
今後 3 年をかけて社員が一丸となって未来の組織像を実現していく。

<キーワード>

共創アプローチ、社員参画型、経営計画立案、自分事化、関係性分析、モデル化、構造化

<想定する聴衆>

経営者、上級管理層

<活動時期>

2016 年 5 月～2017 年 3 月末（継続中）

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
■ 改善活動を実施したが、結果はまだ明確ではない段階
□ 改善活動の結果が明確になっている段階
□ その他（　　）

<発表内容>

1.背景

弊社のこれまでの経営計画立案・展開は、主に経営企画部門と役員クラスの事前調整に基づくトップダウンアプローチであった．実務層は実施すべき施策が指定された全社計画に合わせて部門・部署計画を立案、展開していた．結果的にやらせる／やらされる運営が中心となり、躍動感やワクワク感のない、当事者意識・参画意識が薄い、“こなす”仕事が多いと感じていた．また、全社レベルの計画の意図、展開内容と実務レベルの実質的な実践内容が乖離している場合も存在した．

2.改善したいこと

これまでのトップダウンアプローチに偏った経営計画立案・展開～実務層におけるやらせる／やらされる運営から脱却し、社員一人ひとりが当事者となり事業を進めることで“仕事を通じた幸せ”を社員が獲得する中期経営計画の策定を目指した。

3.改善策を導き出した経緯

2016 年に新社長が就任し、創立 50 周年(2014 年)に掲げられた新テーマ“IT で幸せに挑む” の実践・実現に向け、中期経営計画立案を進めることとなった．

中期経営計画立案を進めるにあたり、その命を受けた経営企画部門の主担当者が、これまで社内外問わず、関係者が自ら現状分析～改善計画立案とその実践を当事者として取り組む場を作りファシリテートするサービス（SaPID(*1)）を提供してきた筆者に（2016 年 5 月初旬）支援依頼してきたことから当取り組みが動き始めた．

次期中期経営計画発行予定時期が 2016.12～2017.1 頃と期間的な余裕があること、実態に即した実効性のある計画にしたいこと、社員との直接的なコミュニケーションを重視する方針であった等の理由から、次の 1)～4) の情報を可能な限り役員～実務者より直接収集し、事実情報を分析することにより、弊社の実情や社員の想い、そして社長を中心としたトップマネジメントの想いや考えを融合させ、実効性ある（段階的な）施策として反映した中期経営計画の立案方法を検討した．

1)弊社らしさや実務で大事にしていること
2)こういう会社になってほしいという将来像
3)弊社の強み：他社より秀でた点、ウリ
4)弊社の弱み：実存する課題や問題、困り事

これらの事項を“モノゴトを自分事にするための影響要因”に着目した社員参画型共創アプローチとして設計し、提案した．“共創アプローチ”とは、トップから実務を行う社員までが一体となり、共に新しい組織や事業を創りあげ運営することを促進する施策群、という意図で命名した．

4.改善策の内容

社員参画型共創アプローチ（改善策）として、以下の施策を実施した．

（1）社員による関連情報収集と分析

多くの社員からの意見、コメント収集の手段は、主にワークショップ形式を採用した。参加者に各 5 人前後のチームになってもらい、3.に示した 1)～4)それぞれに対する各自の意見、コメントを付箋に記述し、その内容をきっかけにして相互に対話しながら本当に言いたいこと、伝えたいこと、背後に存在する事象や問題・課題などを明らかにした．それぞ

れの社員からの意見、コメント収集やその結果共有による Q&A の実施実績は以下の通りである.

表 1：社員の意見、コメント収集～結果共有活動実績

実施日 2016 年	実施内容	参画者数
5 月初旬	新入社員ワークショップ	25
5/18	新任課長代理ワークショップ	16
6/20	管理職選抜ワークショップ	17
7/8	新任主任ワークショップ	22
7/15	串刺懇談会ワークショップ	7
8/3	経営管理部門管理者ワークショップ	10
8 月末	経営管理部門ワークショップ	40
9/1	部門長ワークショップ	16
9/12-13	役員合宿ワークショップ（全員出席）	12
9/29	データセンター部門ワークショップ	15
11/25	技術部門管理者ワークショップ	11
11/26	公共部門管理者ワークショップ	15
12/6-7	技術部門意見交換会	15
12-1 末	社長による Town Meeting(計 5 回)	224
		計 445 名／800 名（56%） 課長以上の管理層参画率＝80%超

当初は筆者がファシリテータとなりワークショップを運営した．抽象度が高い内容、事実かどうか不明な事項、真意が不明の表現などを“質問を契機にした対話”により掘り下げ、周囲のメンバー（他者）にも理解できる具体的な内容として書き換えて共有した.

主な付箋（要素）の処理（洗練）が終わったところで、それぞれのチーム、もしくは参加者全員で収集（列挙）した個別要素を分類し束ねる、表題を付与する、因果関係が成り立つ要素間を矢印で結ぶ（原因→結果）ことによるモデル化・構造化を行った。これを関係性分析と呼ぶ．（図 1）

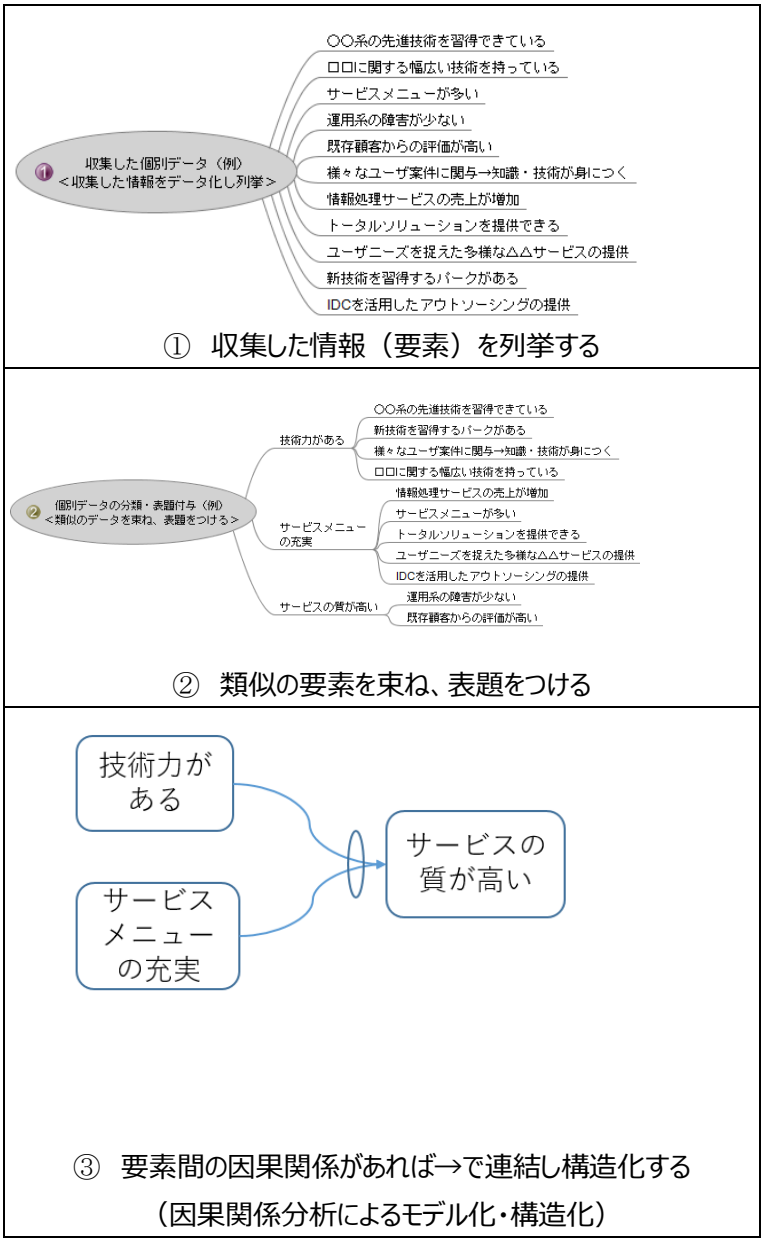


図 1：関係性分析の流れ(例)

分析対象とその成果物は以下である.

・1)＋2)から　弊社の特長を統合したこういう会社にしたいよねモデル
・3)＋4)から　わが社のよい点＋問題・課題構造図

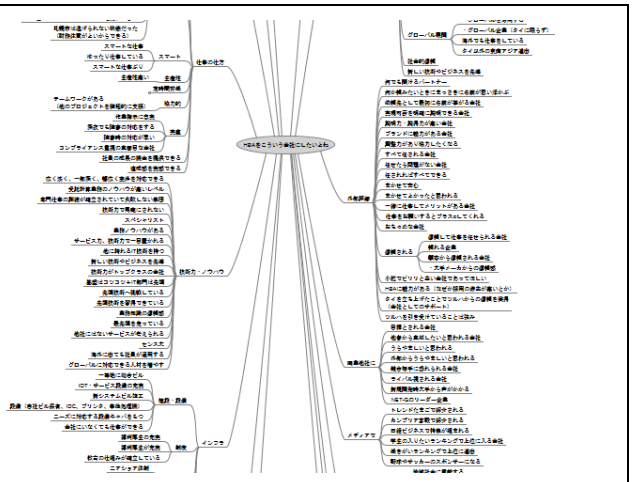
ファシリテータは、要素の洗練や分析、モデル化・構造化に対しての答えを提供するのではなく、これらの進行促進と答えの出し方をガイドする.見落としや異なる見方などのヒントを提供することで、参加者自らが要素を洗練し、構造化した結果を導き出すことを促進する. 参加者自らが記述した要素を、自ら分析し、構造化することで、メンバー全員の総意である結果が導出されるため、参加者全員が腹落ちしやすく、自分事化しやすい結果を獲得することができる.

なお、筆者がファシリテータを自ら務めたのは当初から 2016/9/1 に実施した部門長ワークショップまである. それまでのワークショップにて進行方法やファシリテートの仕方を確立し、主催する経営企画部門に技術移転し終えた. 以降のワークショップ運営、ファシリテートは経営企画部門に委ね、筆者は次の段階となる収集済み全データの分析・構造化に

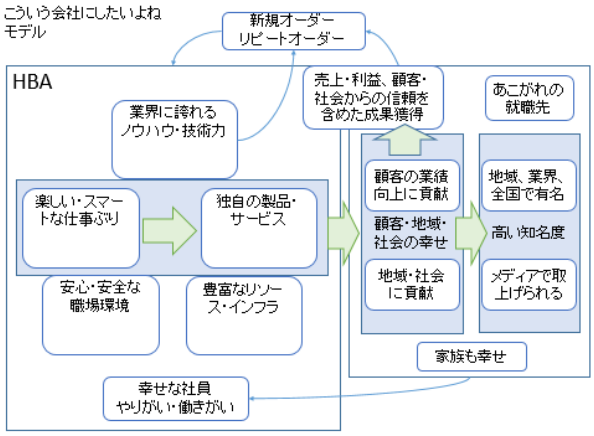
軸足を移した.

（2）収集した全データによるモデル化・構造化

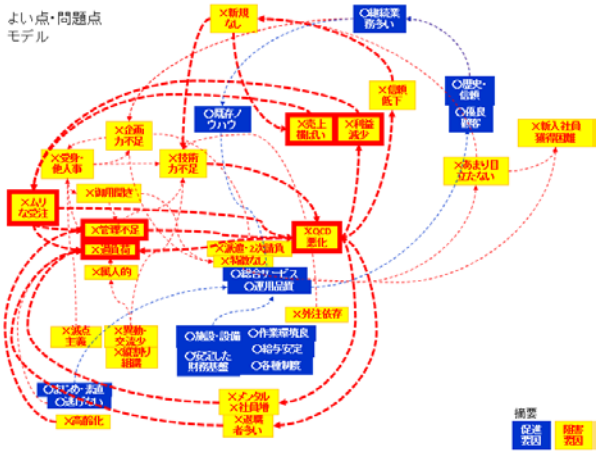
（1）で個別ワークショップごとの、こういう会社にしたいよねモデル、および、わが社のよい点＋問題・課題構造図を構築したが、それらは“参加者層”から見えている限定した情報に基づくものである. 情報の抜け漏れ、偏りが存在し、全社経営方針・計画立案の入力情報として相応しくない. よって中期経営計画立案のために全データによるモデル化・構造化が必要との判断し、それぞれのワークショップにより収集した 1)～4)の情報（要素）をすべて電子化し、関係性分析を行った.



② 収集した情報のデータ化＆分類（一部）



③ 会社の特長とこういう会社にしたい要素の因果関係分析から構築した「こういう会社にしたいよねモデル」



④ わが社のよい点＋問題・課題要素の因果関係分析から構築した「よい点＋問題・課題構造図」

図 2：こういう会社にしたいよねモデル／
よい点＋問題・課題構造図の構築

（3）解決すべき問題・課題（候補）の定義

収集したさまざまな情報（要素）の洗練は（ワークショップの時間・工数的な制約から）ファシリテーション過程で検出した要素に絞って対応したため、分析結果と上記モデルには解決手段の裏返し事項などが残存していた.
例えば「～不足」（例：プロジェクトマネージャ不足）がそれにあたる. この情報は「～（例：プロジェクトマネージャ）を充足させると（ある問題が）解決する」という意図を持った内容であり、解決したい問題・課題そのものではなく解決手段の提案となっている. この情報をそのままにすると、本当に解決したい問題・課題に対する解決手段の他の選択肢を排除してしまい、実効性のある施策を導出しにくい状態となる.

よって、この段階で解決すべき問題・課題は何か？を再整理し、解決すべき問題・課題の候補として定義することとした. （例：ロスコンプロジェクト化の防止）

（4）中期経営計画テーマの再確認

限られたリソースや制約条件の中で、最も有効な施策を実践するためには、沢山の解決すべき問題・課題の候補群の中から選りすぐった問題・課題を特定し、限りあるリソースを集中投下する必要がある.
この段階でその判断基準となる中期経営計画のテーマを今一度再確認した.

（5）社長主催タウンミーティングの実施

以上の情報収集、分析結果から社長、役員、および経営企画部門が中期経営計画の骨子を検討した. その結果を携えて社長自らがタウンミーティング（札幌・東京にて計 5 回）を開催し、ワークショップに参加できなかった社員などを中心にこれまでの取り組みの経緯や情報収集・分析結果、今後の方向性（中期経営計画の考え方など）などを説明した.
また、説明後には参加者からの質問一つひとつに社長が自らの考えを伝達し、全社員が一丸となって社員・顧客、そして会社が幸せになるために一緒に取り組む意思を共有した.

（6）中期経営方針と中期経営計画の立案

以上のように収集した情報（要素）を分析・構造化し、解決すべき問題候補と中期経営計画のテーマに基づいて、解決すべき問題・課題を特定し、役員と経営企画部門が中期経営方針を策定、中期経営計画を立案した.

5.改善策の実現方法

今回の改善策の実施において、以下の点に注意した.

（1）社員の参画・自分事化を促進

「実務の自分事化」「仕事を通じた幸せの獲得」「幸せな社員が顧客を幸せにする」を目指す上で、社員の参画は不可欠である. その意味で、アンケート収集などの間接的な手段ではなく、あえて直接対話の機会となる（社員間交流も促すことを目指して）ワークショップやワールドカフェ形式（Town Meeting で実施）を主な手段として採用した.
自らの表現で記載した付箋や他者が記載した付箋を一緒に参画した社員たちと相互に理解し、一緒に全体像を創りあげる.
構成要素、全体像の両面に納得し、合意を形成する.
以上の過程を踏むことで成果物の内容に対する自分事化を促進する.
また、可能な限り定例開催される昇格者向け社内研修の場を活用するなどして社員への負担の軽減を考慮して実施した.

（2）安全な場の提供

テーマに対して参加者が忌憚のない意見、コメントが出せるように安心、安全な場を提供することが重要である. そのような場を提供するために実施した事項を列挙する.

・役員、管理者、実務者のワークの分離
・チームメンバーはあらかじめ運営側で設定することで、不仲、不毛な争い等が発生するリスクを最小化する
・情報収集・共有時の匿名性の確保
・ファシリテータ自らぶっちゃけ話を提供する
・感情的な内容であってもその背景を冷静に掘り下げ、その真意や背後に隠れている問題を把握する

（3）事実情報の収集

実存しない情報を基に計画を立案するようなことはあってはならない. しかし、多くの社員から意見・コメントを収集すると事実かどうか把握できないもの、疑わしいものが混在していることが多い.
一方で、収集前に「事実でなければなりません」などの注意を述べてから収集を始めると、とたんに参加者の手が止まり、大事な情報も出てこなくなる傾向がある.

よって、意見・コメント収集時には“テーマ”以外の制約を可能な限り排除した上で記載してもらい（大事な情報を逃さないように）、その内容をヒントに掘り下げる（事実かどうかを把握して実存しない場合は排除する）対応を行った.
また、“問題・課題”という表現が硬い、難しいと受け取ると手が止まることを考慮して“普段の実務における困り事”を書いてもらうようにした.

（4）明るい未来とよい点から始める

意見・コメントを収集する順番は、以下のとおりである.

1)弊社らしさや実務で大事にしていること
2)こういう会社になってほしいという将来像

3)弊社の強み：他社より秀でた点、ウリ
4)弊社の弱み：実存する課題や問題点、困り事

計画立案などを目的とした現状把握情報として一般的に収集されるのは問題・課題関連情報である．存在する問題や課題を捉えることでその対策としての計画を立案することが可能である．しかし、問題・課題情報から収集・分析し始めると、現実やマイナス面ばかりに目がいき、参加者の顔が曇り、後ろ向きな．躍動感のない取り組み内容になってしまう．それでは"共に創る：共創"は実現できない．
よって、前向きで発展的な場を作り、新しい組織を一緒に創りあげる、参加者が前を向いて取り組んでもらえるように、1)～3)に関連する情報を先行して収集し、前向きな場を作り上げてから 4)を取り扱うこととした．

（5）自分事化を促すアプローチ
以上のワークは、自分事化影響要因を考慮した対策を、適切なタイミング、必要な箇所に配置し、構成した．（表 2）
この対策群（アプローチ）により、ワークショップの参加者である社員が自らなりたい姿・ありたい姿と現状を把握し、そのギャップを埋めるために必要な施策を自分事として実践することを促進する．

表 2：自分事化影響要因とその対策

自分事化影響要因	今回の対策
自分が言ったこと、決めたことに責任を持ちやすい． 誰かが言ったことは自分事にしにくい．	・ワークショップにて自らの表現で書き出した要素を基に分析・構造化を実践してもらう．
自らのことを周囲に理解してもらいたいと（内心）思っていることが多い→理解してくれるとうれしい． 自分の困り事を解消するためなら積極的になりやすい．	・自分の困り事を書いてもらうところから始める． ・その内容と真意をチームメンバーがリアルに理解するワークを行い、その結果を分析・構造化に活用する．
一人だと継続できないことも仲間とならやれる．	ワークショップではチームとして取り組んでもらう．
価値を感じていれば、自分でもできる/できそうだと思うことには挑戦する．	・自分が考える"こうなりたい姿"を表明してもらうところから始める． ・自分の困り事を書いてもらう． ・なりたい姿と問題構造のギャップを埋めるために、チーム全員で必要な事項を決定（合意）する．
できなかったこと・わからなかったことができる・わかるようになると楽しい．	沢山の問題・課題のどれに取り組めば合理的、現実的なのかを構造化により明らかにする（自ら答えを導く）．
自らのニーズ・要求がよくわかっていない．問題や要求ではなく（自分が知っている）手段を提示してしまう．	実務での困り事、感じていることを書いてもらい、その情報から（適切な質問により）背後に存在する問題・課題を明らかにする．
場に行動が左右されやすい． 問題・課題を表明すると不利益が被る場では、ありのままの状態を誰も話さなくなる．	安心、安全な場を提供し、1)～4)の情報を生の声として獲得する．

6.改善による変化や効果

以上のアプローチにより、2017.4.1 からの 3 か年計画となる中期経営方針・計画を発行し、その実践に移行してい

る．これまでの実践結果と現時点での効果および、今後の課題と対策を記載する．

1）実践結果と現時点での効果

（1）社員の積極的な参画を獲得

＜実施前の不安＞

ワークショップを実施しても冷めた態度を取る社員や管理者が多数存在することになるのではないか．

＜実施結果＞

いざワークショップを開催してみると、ほぼ例外なく参加者がみな積極的に参画し、実践していた．特に意外だったのは、普段は強権発動的な行動・言動が目立つ社員や、一匹狼的なリーダが、ワーク時のチームメンバーと共にバランスのよい真っ当な意見・コメントを出しつつ分析や構造化も積極的に進めていたことである．

普段は内弁慶的に振舞っている社員であっても、同じ役割層等他者の目に晒されること、およびファイシリテータにより極論やバランスを欠くコメントには別の見方や事実確認を行うため、出過ぎた行動が融和・抑制された可能性がある．このアプローチには極論や自分勝手な筋違いな内容等が除去される効果があるのではないか．一方で、（誰でも自分は正しいと思っているので）自分の意見・コメントが通らないことを心の奥底に溜め込んだ可能性もあるため、今後も継続観察していく必要がある。

（2）長所・強みが悪循環に加担していることに気づいた

＜実施前＞

事業、業務上の好循環、悪循環は、それぞれ長所・強み・よい点／短所・弱み・悪い点だけで構成されていると思っていた．

＜実施結果＞

今回は長所・強み・よい点／短所・弱み・悪い点すべての要素による関係性分析を行った結果（図 2 を参照）、弊社の誰もが認める長所・強みの一部が、事業の悪循環に加担している可能性が把握できた．（表 3）

このことは、今後弊社が目指す姿をあらためて考え直すきっかけとなった．

表 3：強みが悪循環要因へ加担する（例）

例： 強み「トータルサービスを提供している」 →解釈「何でも出来る」 →受取「ぼんやりしていて特長がない」 →結果「新規顧客獲得がなかなか進まない」 ＝悪循環要因の一つ

（3）社員の想いとトップの想いの融合を促進

＜実施前の不安＞

実務層の想いとトップの想いが折り合わない可能性が高いのではないか．

＜実施結果＞

実務を通じて社員が何を見ているのか、どう感じ、何を想っているのかを把握したうえで、トップの想いと融合することができた．これまでも現状評価に基づき計画を立案してきたが、社員の生の声を取り入れ、その背後にある事実を把握した上でトップの想いと明確な形でつなげることができたのは初めてのことである．

今後「実務の自分事化」「仕事を通じた幸せの獲得」「幸せな社員が顧客を幸せにする」を目指す上で大事な階段を一つ昇ったと言えるのではないか．

このような結果となったのはこの取り組みが、「（仕事を通じた）社員の幸せ」「顧客の幸せ」そして「会社の幸せ」の実現（三方よし）を目指しているからではないかと思う.

（4）解決すべき問題・課題（候補）の明確化による実効性ある施策導出

＜実施前の状態＞

これまでの中期経営計画、年度事業計画で打ち出されてきた施策は、弊社の（そのときどきの）現状に対して間違いではないものの抽象的、一般的な内容であったり、表現は異なるものの何年経っても同類の内容が繰り返されることが多かった. そして実務層や管理層からの生の声を反映したとは言いがたい状態であったため、施策に対する社員の当事者意識は薄く、実務とは別物的に扱われている印象もあった.

＜実施結果＞

今回のアプローチでは、実務層・管理層から直接声が上がった内容を基に、解決すべき問題として捉えなおし定義した（→4.(3)参照）候補群から、中期経営方針に沿って選りすぐった事項に対して施策を打ち出されている. 仮に、これまでと同じ（同類の）施策が存在していたとしても、自分事として捉えて取り組む可能性が高いと言える. 現在、各部門・部署の中期経営計画立案中であるため、今後の取り組みで自分事としての実践をさらに促進し、その成果を確認する必要がある.

2）今後の課題と対策

（1）全社事業計画はどの部門にも適用可能な（抽象度を高くした表現の）施策として提示しているため組織の状況や特長に適応できるアプローチを設計する必要がある.

＜対策＞

各部門による施策具体化検討開始時に、いま一度各種分析結果、施策の目的、背景、現状、体制、制約条件などを共有し、効果的かつ現実的なアプローチを設計してもらう. そのうえで小さく試行した結果を活用してから部門内展開につなげるよう促す.

（2）人間の感情論をも取り扱うため、比較的高度なファシリテーション技術が必要になる.

＜対策＞

有識者が先行実施しながら、進行方法やファシリテートの仕方を確立し、ファシリテータ候補者に技術移転する.（→4.(1)参照）

（3）人の心を動かすのは簡単ではないため、継続して手間／時間をかけて対話していく必要がある.

＜対策＞

即効性のある手法は存在しない. わかってもらえるまで粘り強く、一貫して対話を重ね、以降もさまざまな局面で繰り返しミッション・ビジョン・価値観を伝え、共有し、その実現を目指す実践を継続していく.

7.改善活動の妥当性確認

2017 年 4 月より中期経営計画の実践に入り、当活動の妥当性確認を行う.

しかし市場環境の変化は激しいため、計画した施策を実践するだけではなく、併せて中期計画に対する四半期単位のふりかえりと計画見直しを実践していく.

また、今後の実践課程で社員が仕事を通じた幸せを実感できるようになったのかを（2011 年から隔年で実施している）社員満足度（ES）調査の幸福 5 指標（心身の健康・経済的安定・良好な人間関係・仕事への情熱・周囲への貢献）などの指標を活用して評価していく予定である.

A. 参考情報

[1] ソフトウェアプロセス改善カンファレンス 2012（SPI Japan 2012）システムズアプローチに基づくプロセス改善メソッド：SaPID が意図するコト（Systems analysis／Systems approach based Process Improvement methoD）～プロセスモデルをより有効活用するために／そして現場の自律改善運営を促進するために～
http://www.jaspic.org/event/2012/SPIJapan/session3A/3A4_ID023.pdf

[2] SaPID 実践事例より～改善推進役がやるべきこと／やってはいけないこと
現場が自らの一歩を踏み出すために
http://www.jaspic.org/event/2013/SPIJapan/session2B/2B3_ID011.pdf

3B2「GQM を拡張した WG 活動の短期化」山邊人美（住友電工情報システム）

<タイトル>

GQM を拡張した WG 活動の短期化

<サブタイトル>

なし

<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）： 中村 伸裕（なかむら のぶひろ）
所属： 住友電工情報システム株式会社 QCD 改善推進部

<主張したい点>

当組織では、GQM を活用した WG 活動を実践しており、一定の効果は得られた。しかし組織目標を達成するためには、WG 活動のさらなるスピードアップが重要課題であった。今回の発表では、一定期間で WG 活動の効果を出すために当組織で取り組んだ施策と成果を紹介する。

<キーワード>

GQM、WG 活動、仮説思考、評価報告書、ブランク資料、測定

<想定する聴衆>

改善活動の推進者、改善活動に関わる管理職、改善活動の実施者

＜活動時期＞

2015 年 3 月～2017 年 6 月（現在も継続中）

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

■ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階
□ 改善活動の結果が明確になっている段階
□ その他（改善活動中。 一部成果がでている段階）

<発表内容>

1.背景

当組織は 2017 年の中期計画の目標達成のために、各工程や開発種別(派生/保守)毎に WG を設立し、改善活動を実施している。前回[1]は CMU 版 GQM(*1)を適用し、試行で必要なデータを測定できるようになり、短期間で評価報告書が作成できるプロセスが確立できた。しかし上流の問題分析や課題設定のプロセスでは依然時間がかかっていた。2017 年の中期目標を達成するには、さらなる WG 活動のスピードアップが必要で、上流のプロセスを含む全てのプロセスにおいて短期間で成果を出す方法を考え出すことが課題であった。

(*1) CMU 版 GQM（Goal-Driven Software Measurement）とは
カーネギーメロン大学・ソフトウエア工学研究所が、オリジナルの GQM（ビクター・バシリ教授が開発した手法）
を拡張したもの。メンタルモデルを活用した要因抽出と Indicator(表示方法)の追加が大きな特徴である。

2.問題分析

時間がかかる原因を調査、分析した結果、以下の問題が検出できた。

(a) 問題分析プロセスで、参加者の全ての意見を採用して全ての調査をしようとしていた
(b) 調査を実施しても課題の特定ができない
(c) 効果の期待できる施策がなかなか思いつかない
(d) GQIM の indicator を必要以上に作成していた
(e) 進捗管理をしていない
(f) その他

以上の問題を解決するために、調査を進めた結果、(a)(b)は仮説思考、(d)はブランク資料を使った手法で解決できると考えた。(e)は進捗管理を強化することにした。これら 3 つの施策を 3 章で詳しく説明する。

3.実施した施策

3.1.仮説思考[2]の導入

(1) 現状の問題

当組織では、最初に現状調査を行い、問題分析、課題設定という流れで進めているが、問題分析プロセスで、参加者の全ての意見を採用して全ての調査しているため、時間がかかっている(a)。問題分析プロセスでたくさんの調査結果が出てくるので、課題特定プロセスでは、検討すべき要因が多い。そのため、それぞれの要因の議論に十分な時間をかけることができず、十分な掘り下げができない。掘り下げが浅いので効果が期待できる課題が特定できない(b)。
一方、業務経験が豊富なメンバーは、過去の経験や知識から、どこに問題があるか、どこを改善すれば効果的であるかをわかっていることがある。文献[2]を参考にして、経験や知識から想定した仮説を立てて調査や分析の対象を絞り込み、時間短縮することにした。

(2) 施策内容

仮説思考とは、物事を答えから発想し、それを分析して証明するアプローチである。一般的に、情報は多ければ多いほど間違いのない良い意思決定ができると思われているが、情報を網羅的に調べてから答えを出すことは、時間的にも資源的も無理がある。仮説はまだ十分な情報が集まっていない段階、あるいは分析が進んでいない段階での自分なりの答えである。

問題解決のスピードアップを目指すことができる手法であると考え、WG メンバーに仮説を立てることを推奨したが、「どう書いたらいいかわからない」との声が多かった。そこで WG に参画して一緒に仮説を考えた。最初に作成した仮説をサンプルとして標準に掲載すると、他の WG も記述できるようになった。以下は IT 工数削減の仮説例である。

■IT 工数削減の仮説例

IT に流出している欠陥の XX%は単体テストで検出できるもので、IT に流出する欠陥数を減らせば ・IT における単体テストレベルのテスト項目が削減でき、 ・IT の回転数が減り ・欠陥の記録等の作業が減り IT 工数の XX%が削減できる。

仮説には、原因とその原因がどの程度発生しているかの想定、達成すべき状態、効果の想定を記述することにした。このように記述することで、WG 活動が失敗しても仮説のどの部分がどう間違っていたのか検証することができる。また前提となる XX%の想定が間違っていたのか、施策が悪かったのか、効果が楽観的すぎたのかが判断できる。このような仮説を WG メンバーで合意できれば、問題分析のポイントや施策の検討範囲を絞り込むことができ、効率的に WG 活動を進めることができる。ある程度仮説が出てきた時点で、WG メンバーが最も効果が期待できる仮説を選択する。

表 1 はコスト削減の仮説に対して、それぞれどの程度コスト削減できるか WG メンバーの主張をまとめた表である。このような表を作成することで、個人的な思い込みを排除することができ、また仮説を納得感を持って選択することができる。

表 1.コスト削減の仮説に対するメンバーの主張

仮説	A さん	B さん	C さん	D さん	E さん	実施
仮説 A	20%	10%	0%	0%	25%	
仮説 B	10%	0%	5%	0%	10%	
仮説 C	30%	15%	20%	25%	15%	○

仮説によって課題を決めるアプローチは、短時間で課題を設定できるというメリットがあるが、事実とは異なる固定観念(思い込み)であることも多い。仮説の信頼度を高めるために、以下を確認することにした。

・ 仮説に含まれる原因の具体例を 2 ～ 3 集め、仮説が正しそうであることを確認する。

・ ビジネスゴール、仮説、課題(achieve)、仮定、施策が論理的につながっていることを確認する。

これらの関係を表したものを図 1 に示す。仮説は課題が達成できれば、ビジネスゴールが達成できる説明であり、仮定は施策を実行すれば課題が達成できる根拠である。

図 1. ビジネスゴール、仮説、課題、仮定、施策の関係

図 2 に仮説思考を導入した当組織の WG 活動プロセスのフローを示す。仮説設定を追加することで、問題分析の調査や分析

の作業量を減らすことができ、期間を短縮できる。

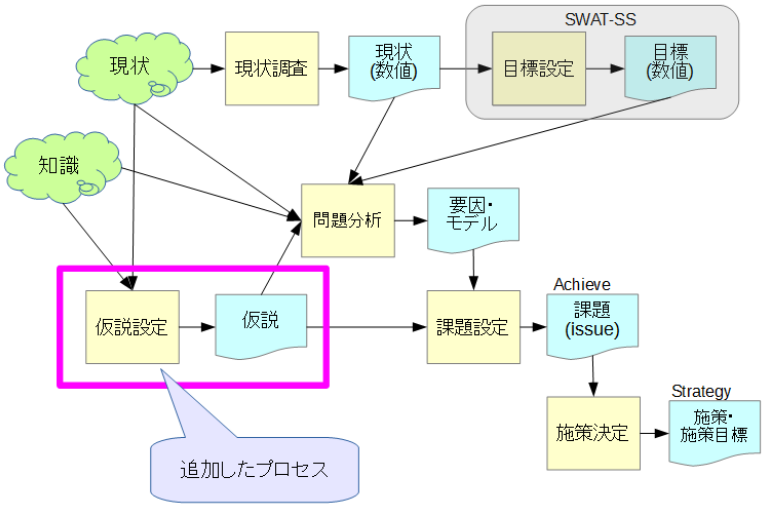


図 2. 現状調査～施策決定の流れ

(3) 成果

仮説思考の効果を WG 活動開始から 3 ヶ月後に課題設定できた WG の数で評価することにした。その結果を表 2 に示す。WG 数は今年度新しいテーマで取り組んだもののみカウントしている。課題設定できた WG の割合は 8%から 71%に向上し、仮説思考の効果はあったと思われる。

表 2. WG の課題設定が 3 ヶ月で達成した割合

	WG 数	課題設定(3 ヶ月後)	割合
2016 年度	12	1	8%
2017 年度	7	5	71%

また、施策の仮定を書き出すことで、課題や施策の不整合に気付き、軌道修正した WG があった。早い段階で気づけたことで、大きなロスは発生せず、時間短縮に貢献できたと考えている。

3.2.ブランク資料作成プロセスの導入

(1) 現状の問題

当組織では、施策決定し試行を実施する前に、評価のための測定計画を立案している。目的に沿った測定をするために GQIM Tree(*2)を取り入れているが、評価報告で使用する indicator の 1.5～2 倍くらいの indicator を作成していることがわかった。WG メンバーが提案した indicator を全て採用していたり、同じ測定ゴールに対応する indicator が複数定義されていたりして、検討期間が 2 ～ 3 ヶ月に及ぶ WG もあった。

(*2) GQIM Tree とは

GQIM(Goal:測定ゴール Question：質問 Indicator：インジケータ Metric：メトリクス)の関係をツリー形式で表現したもの。測定ゴール、質問を定義することで、何のために測るのかといった測定目的を常に意識することができ、目的に沿った意味のある測定ができる。また、インジケータとその解釈方法を合意することで、評価が恣意的になることを防ぎ、そのインジケータを構成するメトリクス(測定方法・測定尺度)を定義することで、評価の段階で、データ不足、データ不備等による測定不可能という事態を防ぐ。

(2) 施策内容

期間を短縮するためには、作成する indicator を絞り込む必要があった。一方、コンサルタント会社では提案資料を効率的に作成するために、ブランク資料を作成している[3][4]。ブランク資料とは、具体的なアウトプットイメージを持つために作られる資料で、目次、主張、主張の根拠を示す表やグラフなどで構成されている。実際の数値が入っていないため(測定前なので)、ブランク(＝空(カラ))資料と呼ばれている。ブランク資料を作成し、主張の根拠を示すためにどんな表やグラフを使うと効果的なのかを議論してから作業を開始することで、不要な作業を防ぐ効果がある。

当組織では、ブランク資料を WG 活動の評価報告書に応用し、indicator の絞り込みに活用することにした。評価報告書のブランク資料を作成することで、報告を受ける人の視点で考えることができ、不要な indicator を削減できる。

図 3 にブランク資料作成を導入した WG 活動プロセスのフローを以下に示す。評価報告書ブランク資料を作成するプロセスを追加することで、不要な Indicator が削減できる。GQIM Tree を作成するプロセスでは、ブランク資料の indicator を転記する手順にすることで、GQIM Tree の indicator を減らすことができる。

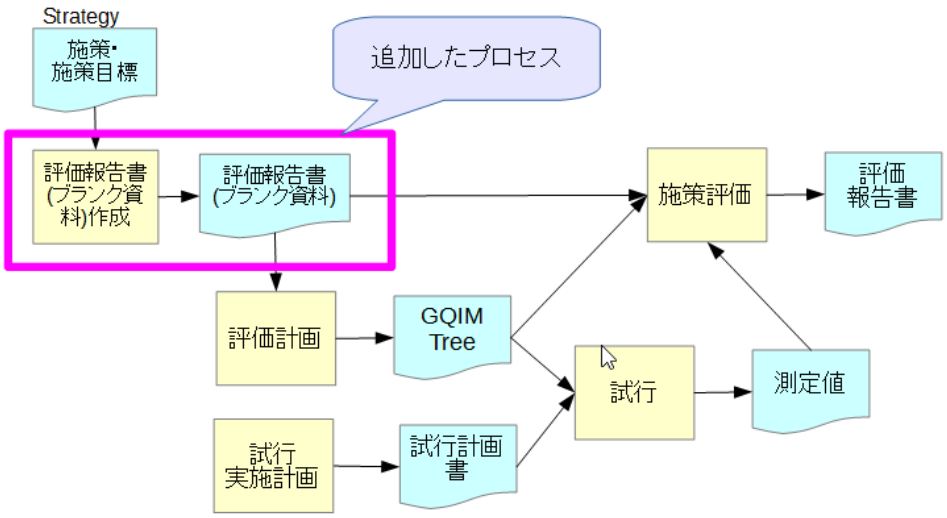


図 3. 評価計画～試行、評価、報告の流れ

コンサルタント会社で活用されているブランク資料は作成の難易度が高いため、トレーニングを実施することにした。トレーニングでは、(a)主張を証明する indicator が作成できることと (b)チームで最適な indicator に絞り込めることを目標とした。まず、(a)を習得するために、個人でブランク資料を作成して、GQIM Tree を作成する という演習とした。次に (b)を習得するために、チームで議論するための演習を設定した。1 チーム 3 名に構成し、全員が積極的に意見を出せるようにした。

また、WG が効率的に作業できるように、評価報告書を標準化しテンプレートとサンプルを提供した。

(3) 成果

ブランク資料と GQIM Tree 作成のトレーニングを以下の通り実施した。

表 3.トレーニング実績

実施回数	3 回（1 回 2H）
受講者数	36 名

トレーニング受講者のアンケート結果を図４、図５に示す。習得度は、①アドバイス・指導できる②一人でできる③WG メンバーと一緒ならできる④支援があればできる⑤作成できそうにない の回答選択肢から回答いただいた。7 割弱の受講者が 1 人で、

または WG メンバーと一緒ならできるとの回答で、図 3 のプロセスが実施可能になった。

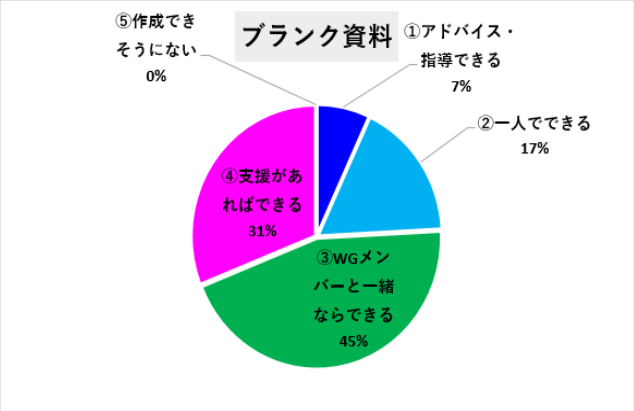


図 4. ブランク資料の習得度

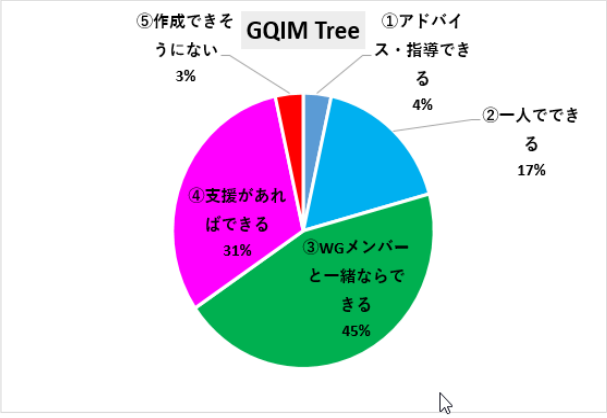


図 5.GQIM Tree の習得度

現在のところ、評価計画プロセスに着手している WG はないため、ブランク資料の効果はまだ測定できていない。10 月の発表までに効果がわかる見込みである。

3.3 標準日程の提供

(1) 現状の問題

WG 活動に時間がかかるその他の原因として、計画的に進められないということがあった。そもそも WG 活動は、各プロセスの標準的な期間がわからない(基準値がない)ので、WG 活動の計画が立案できずにいた。また WG 活動は明確なリーダを設けない(司会を輪番制で行う)という運用方法で実施しているため、コントロールをする人が明確になっておらず、WG 活動の進捗状況は分からない状態であった。

(2) 施策内容

改善計画が立案できるように、必要なタスクを洗い出し、基準値を決めて、標準大日程計画のテンプレートを作成した。図 6 に例を示す。WG 会議は月 2～4 回開催されるので、1 ヶ月単位で進捗を確認できるようにした。
WG 活動をコントロールするために、責任者を決め、大日程計画を使い、月次のトレース会議で報告するようルール化した。

(3) 成果

標準日程を提供することで、WG 活動の進捗状況がわかり、遅れている場合は是正するため、ほぼ計画どおりに進める WG が増えた。

No.	タスク	Schedule (2017-01～2017-12)	開始予定日 開始日	納期 完了日
■目標設定				
1	品質大会での発表			2017-01-15
2	WG改善活動キックオフ			2017-01-31
■現状調査・問題分析				
3	現状調査		2017-01-05	2017-01-31
4	問題分析		2017-02-01	2017-02-28
5	課題特定		2017-02-01 2017-03-17	2017-02-28 2017-05-08
■施策検討				
6	施策のアイデア出し		2017-03-01 2017-03-17	2017-03-15 2017-05-08
7	仮定の抽出と評価		2017-03-15 2017-03-17	2017-04-15 2017-05-08
8	施策の優先付け		2017-03-15 2017-04-21	2017-04-15 2017-05-08
9	シミュレーション		2017-04-15 2017-04-21	2017-04-30 2017-05-08
■試行				
10	評価報告書(ブランク資料)作成		2017-05-01 2017-05-08	2017-05-31
11	評価用GQIM Tree 作成		2017-05-01 2017-05-08	2017-05-31
12	試行の実施計画		2017-05-01 2017-05-08	2017-05-31
13	試行準備		2017-06-01 2017-05-08	2017-06-30
14	試行		2017-07-01	2017-10-31
15	試行状況の確認		2017-07-01	2017-10-31
16	施策の改良		2017-08-01	2017-11-30
■評価・報告				
17	評価報告書作成		2017-11-01	2017-11-15
18	WG活動報告会の準備(活動振り返り)		2017-12-01	2017-12-15
19	WG活動報告会		2017-12-20	2017-12-20

＊ 2017-05-19現在

図 6. WG の大日程計画例

4.まとめ

さらなら時間短縮を目指して、仮説思考の導入、ブランク資料作成プロセスの導入、WG 活動の進捗管理を実施した。仮説思考は、時間短縮ができただけでなく、問題の要因を絞り込んで深く分析するため、課題の質も向上したと考えている。また GQM+strategies の仮定による評価も合わせて実施したことで、施策が失敗であることに早く気付くことができるため、短い PDCA サイクルを回すことができるようになった。ブランク資料作成プロセスの導入は、不要な indicator を作成しないという効果だけでなく、事前に indicator が主張を証明できるかを検討するため、評価報告書の質も向上できると考えている。今回の取組の内、半分は結果が確認できていない状態であるが、成果が出なかった場合は、この施策がうまくいかなかった原因を分析し、残りの WG 活動に活かせるようにしたいと考えている。

A. 参考情報

[1] 山邊 人美,”プロセス改善活動における GQM の評価”,SPI Japan 2015, 2015

[2] 内田 和成,”仮説思考”,東洋経済新報社, 2006

[3] 伊賀 泰代,“生産性”,ダイヤモンド社, 2016

[4] 安宅 和人,”イシューからはじめよ”,英治出版株式会社,2010

3B3「「かんばん」から始まった改善と制度化のジレンマ」竹縄俊政（日新システムズ）

<タイトル>

「かんばん」から始まった改善と制度化のジレンマ

<サブタイトル>

～「やらされ感」から自己組織化へ～

<発表者>

氏名（ふりがな）：竹縄　俊政（たけなわ　としまさ）
所属：　株式会社日新システムズ　インダストリアルソリューション事業部

<共同執筆者>

氏名（ふりがな）：　永田　敦（ながた　あつし）
所属：　株式会社日新システムズ　品質保証部

<主張したい点>

プロセス改善は現場が主体となり実施することが重要になる。
約１年半前に「SEPG」に任命され、事業部のプロセス改善活動を実施していった。
現場の問題を抽出し、改善方法を提案、取組を実施していき、実際に現場に実施してもらう段階で
現場に「やらされ感」を感じさせてしまう結果となった。
本発表ではプロセス改善活動の現場への展開の難しさをご紹介する。

<キーワード>

プロセス改善、SEPG、やらされ感

<想定する聴衆>

プロセス改善活動に取り組まれている方、SEPG、SQA
プロジェクトリーダ／マネージャ、管理者

＜活動時期＞

2015 年 10 月～現在推進中

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
■ 改善活動を実施したが、結果はまだ明確ではない段階
□ 改善活動の結果が明確になっている段階
□ その他（　　）

<発表内容>

1.背景

日新システムズでは、会社のさらなる成長を目的とした組織改革の一環として、2015 年 10 月より、全社の改善を推進するための「未来戦略室」の設立と、より現場に密着した改善活動を推進するための「事業部 SEPG」が任命された。

私自身は「インダストリアルソリューション事業部」（以降 ISD）に所属しており、開発担当と兼務で SEPG という役割になり、新米 SEPG として、改善活動に取り組むこととなった。

以下、これまでの課題と改善の取組みを、順を追って説明する。

【Step.0 改善開始前】

ISD における課題の一つは、最近、要件がすべて決まらない状態で開発を開始せざるを得ない案件が増えてきているが、事業部としては、従来のウォーターフォール型での開発が中心のため、このような案件についての開発ノウハウがないことから、たびたび納期遅れや品質問題が発生することである。

【Step.1 アジャイルやろう！】

そんな中、より適切な開発手法を模索していた現場メンバから「アジャイルでやってみたい！」との要望があがった。そこで、未来戦略室のアジャイルコンサルタントのサポートを受けながら、アジャイルプラクティスの一部（インセプションデッキによる目標設定、要求分析からのプロダクトバックログの作成、朝会＋かんばんでのプロジェクト運用等）を取り入れて開発が始まった。要求をエンドユーザ視点でもう一度分析し直し、価値の観点を活用しながら全体設計を実施、要件と設計をつなげた上でのプロダクトバックログでの計画、スプリントバックログで計画を全員で詳細化し、かんばんで見える化しながらのプロジェクト運営、スプリントごとのふりかえりの実施などで、これまでのウォーターフォール型のトップダウンでのプロジェクト運営から、メンバ全員が価値を把握した上で自律的にプロジェクトを進めていくことの重要性を自分たち自身で感じることができた。

結果、プロジェクトは成功し、顧客の信頼を得、ビジネスとして継続するに至っている。

ただし、この時点では意識高い系エンジニアがいる一つのプロジェクトでの改善にとどまっている段階である。

【Step.2 それいい！トップ判断が事業部を変える】

特に、事業部のマネジメント層が注目したのは、「かんばん」による見える化である。アジャイルにチャレンジしたメンバは、アジャイルコンサルタントのアドバイスから、Web ツールなどデジタルに頼らず、職場の壁＋ポストイットを使ったかんばんと、模造紙による手書きのバーンダウンチャートという運用方法でアナログなアプローチを実践したため、必然的にマネジメント層や他プロジェクトのメンバからも状況はわかりやすく、問題の早期発見/早期解決に有効な手段だと評価された。また、これまでおとなしい感じだと思われていたメンバや、協力会社からのエンジニアが、役職や所属の壁を意識せず、毎日かんばんの前で活発にディスカッションしている姿をみて、コミュニケーションが大きく変化したことも評価された。

その結果、マネジメント層は事業部全体・全プロジェクトへの展開を検討し、そこで、３つのプロジェクトで「かんばん」を運用する方針を定めた。つまり、一つのプロジェクトでの改善活動の「一部（アジャイルの展開ではなく、かんばんの展開）」が、一気に事業部全体の仕組みとして採用されたわけである。

運用が始まってみると、各プロジェクトで自律的に自分たちが使用しやすいように工夫（メンバ別かんばん、バグかんばん等）が行われ、自律的に改善する意識が生まれていた。ただ、その工夫は、それぞれのプロジェクト内に留まり、他のメンバに広めるところまではできない状態だった。

【Step.3 自律改善はつなげなければ！】

その後、プロジェクト状況の見える化が一定の評価を得て、これらの先行プロジェクトの「かんばん」をベースに、それぞれの工夫を持ち寄り、事業部全体の「プロジェクトの見える化施策」として、標準的なルールを策定し、さらに定着を加速するため、SEPG に取りまとめとルールの策定が依頼された。

３つのプロジェクトの「かんばん」は、それぞれのプロジェクトによって独自のルールのもとで運用されていた。
そのため、事業部の標準的なルールとして策定するにあたり、

- プロジェクト独自のルールを事業部全体に、どこまで反映するべきか？
- かんばんを運用してよかったこと、悪かったことはなにか？
- より改善することはあるか？

を念頭に、実際にかんばんを運用したプロジェクトのメンバから意見を収集することにした。
その際に、メンバと管理者は視点が違うため、メンバにはメンバ目線で、管理者には管理者目線で以下の質問を投げかけ、意見を集めた。

- ✓ 「かんばんを使って何が良かったか？」
- ✓ 「何が悪かったか？」
- ✓ 「どうしたらよかったか？」
- ✓ 「今後も使いたいか？」

集めた意見の中でも、メンバ、管理者ともに今後も使いたいとの前向きな意見が多く得られたことはうれしく思った。
意見を集積したうえで、プロジェクト特有の問題のためのルールなどは削除し、それ以外の流用できる部分及び、悪かった点について、検討しルール化し、全ての技術メンバへの説明を実施した。
その際に、メンバの同意の上、毎月運用状態を確認し、問題あれば随時ルールを変えていくことも運用ルールとした。
4 月に事業部内に展開説明を実施し、標準的なかんばんルールとして制定した。

ここまでは、順調に進んだ、「ボトムアップの改善をきっかけにした、組織全体への改善の定着活動」は、新たな問題の引き金を生むことになる。

2.改善したいこと
【Step.4　え！そうくる？（メンバの意識の変化）】
　標準的なかんばんルールの制定と導入により、「かんばん」による事業部の見える化が組織全体に展開され、さらなる改善がすすむかと思ったが、かんばんルール制定後、もともと、かんばんを使用していたメンバから、意外な意見をもらった。

1. 「ルール化するとやらされ感満載になってしまった」
2. 「アジャイル開発手法の中から、かんばんだけ抜き出してルール化したような、中途半端なルールはいらない」

という意見である。
それぞれの工夫を全員で共有して、他のプロジェクトの状況の見える化を進めるという目的のためにルール化したのに、このような意見がでたことは非常にショックだった。
とはいえ、批判に聞こえたその意見の奥を考えてみると、つまり、
・ルール化されると自由度が無くなった。ルール以外のことが許されなくなった（1 からの想定）
・かんばんだけでなく、プロジェクト開始時からの流れやスクラム全体の流れも共有してほしい（2 からの想定）
という受け取り方、思いがあると気付いた。
つまり、現場での改善の活動を実施する際、良い成果の横展開と現場への定着による改善の加速のため、よかれと思って実施した「ルール化」というアプローチでは、知らないうちにメンバの心情の中に、「トップダウンからの命令」というイメージが発生してしまい、逆に現場のメンバに「やらされ感」を持たれてしまうことに気づくことができた。

　現場の改善を推進するには、メンバ自身が自分たちで改善策を検討し、進めていくことでができれば、早いスピードで改善が進み、その効果も大きい。実際に、その状況を実感できたのは確かである。しかし、そこに少しでもメンバに「やらされ感」を持たれてしまうと、完全に改善活動が止まってしまう。

メンバに「やらされ感」を感じさせないよう自己組織化を目指すため、まずは私自身の現場へのアプローチ方法について改善することが重要だと考えた。

3.改善策を導き出した経緯

【Step.5　SEPG の目指すべきスタンスとは？】

どうすれば、メンバの心のなかに「やらされ感」を発生させずに改善を引き出し、組織に定着させていくのか。
SEPG としての役割と責任を悩みながら、現場に向き合うための 2 つのスタンスを考えてみた。

そのスタンスが以下の 2 つである。

- 改善することの必要性を感じてもらい、自分たちから改善案を引き出せるようにする
- そのうえで、やらされ感を出さないように現場からのヒアリング方法、アプローチ方法を改善する

ISD メンバが「自分たちで変えている」というイメージを持ってもらえるように、改善意識と改善行動を引き出し、かつ、ルール化などトップダウンに感じてしまうまとめ方をせず、あくまでも自律的な改善として現場に定着させていく必要がある。
その根底には、SEPG として現場からの信頼関係も、これまで以上に強固なものとして再構築する必要もある。

4.改善策の内容

【Step.6　救世主?が新しいアプローチを持ってやってきた！】

2 つのスタンスを活動として具現化するために、SEPG 活動として、メンバの意見を収集する仕組み、その結果を使って現場の自律的な改善意識につなげる仕組みなどを検討し、もっと濃密にコミュニケーションをとる必要性を感じた。つまり、メンバから一方的に意見を聞くだけではなく、一緒になって課題を分析し、共有し、対応策を引き出していくことが重要となる。

では、どのように進めていくべきなのか？を考えている時に、2017 年 5 月から、弊社の品質保証部門にアジャイルでの DevQA などを推進している永田 敦さんが入社するという、一つのきっかけをつかむことができた。

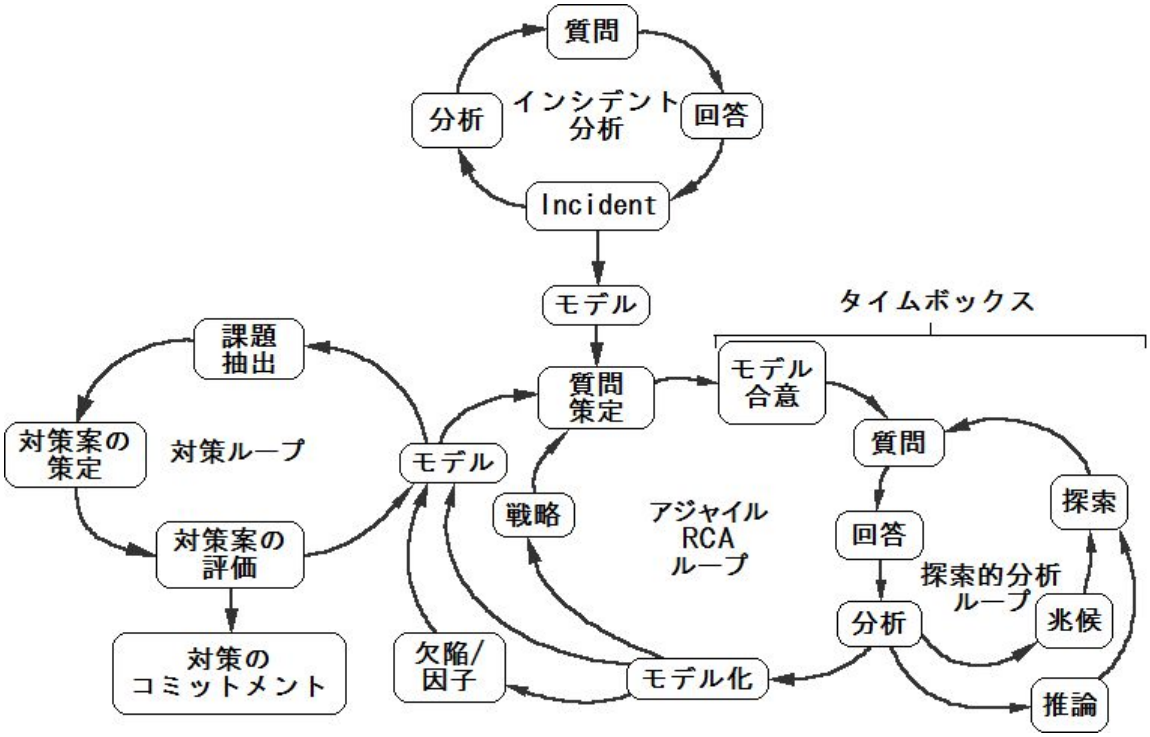
上記の 2 つのスタンスに対して、改善策としては、以下のアプローチを開始している

- アジャイルを適応しているプロジェクトに対しては、アジャイルコーチとして参画し、
  改善の目的説明と具体的な活動のアドバイスによる自律改善の更なる加速（ボトムアップの強化）
- プロジェクトで発生した問題に対して、アジャイル RCA での分析より、真因の自分化（改善必要性の体感）
- 以上の結果から、メンバ自身に対策案を考えてもらえるように SEPG がファシリテートする（自律改善）

5.改善策の実現方法

【Step.7　自律改善の意識向上と信頼関係】

現在 ISD にて、問題のあったプロジェクトに対して、永田さん主導のもと、アジャイル RCA にて分析を実施している。（図：対策まで含めたアジャイル RCA）



図：対策まで含めたアジャイル RCA

具体的には、プロジェクトで発生した障害を使い、30 分の短い時間でのヒアリング→アジャイル RCA での分析・フィードバック→さらに 30 分でのヒアリングというサイクルを回している、内容によっては、3～4 回程度のサイクルで、障害を発生させてしまった真因が表出化することとなる。その多くは、単純なエンジニアリングの問題ではなく、要求分析の不足、派生開発でのプロセス設計ミスなど、大きな問題が背景にあることを引き出せている。
また、SEPG は運営面でのスケジュール調整、ヒアリングの同席、分析結果の解析方法を教えてもらうことで、客観的な視点で気付きを得ることができている。

6.改善による変化や効果

アジャイル RCA の分析内容をメンバにフィードバックすることで、改善すべきポイントが明確になるだけではなく、ヒアリングを通して、メンバは改めて自分たちの問題として再認識することができる。
その結果、SEPG 経由で要求分析のコンサルティングの依頼をするなど、具体的なアクションを自分たちで起こしている。
また、開発のスタート段階からアジャイルコーチによる、アジャイルを進めていく上での狙いやポイントなどをリアルタイムにコーチングしてもらうことにより、ボトムアップで始めたアジャイルでの開発が、さらにブラッシュアップされ、かんばんを組織に展開できた時と同じような「改善のための種」が、また新たに生まれ始めている。
つまり、これらの活動を通じて、改善を進める SEPG や品質保証とメンバとの信頼関係が得られるだけでなく、自ら改善しようという意識が出始めているのである。

SEPG としても、アジャイルコーチやアジャイル RCA の活動の中から、プロセス改善のポイントや、プロジェクトの状況を把握するための SPEG としての観点を得られるだけでなく、信頼関係の構築方法、ファシリテーションの方法など、現場が自律的に改善を進めていく上でのノウハウも獲得できている。
これらの得られた知識やノウハウを、SEPG 自身が整理し体系化することで、どのように活用すれば、ISD メンバが「自分たちで変えてる」というイメージを持ちながら、トップダウンに感じさせない自律的な改善ができるかを考えていくこともできる。

もう一つ、ここの自律改善に対して SEPG からフィードバックすることも重要と考える。
現在実施しているクレーム分析結果など活動の取組と結果については、技術部会で発表するときに、SEPG として今後の動きを説明していくことで、一時的ではなく、継続した改善を実施していくという意識を持ってもらいながら、それぞれの成果を共有することで、組織内での信頼関係を強くしていくことも推進していくことができる。

これらの取り組みによって、改善のやらされ感は低くなり、改善が進んでいく組織に変わっていくと考える。
最終的には、アジャイル、スクラムで自己組織化を目指す。

7.改善活動の妥当性確認

取組中のため、今度の取り組と効果に関しての詳細は 10 月に発表する。

3C1「システムアーキテクトと取り組む品質管理」福良智明（オープンストリーム）

<タイトル>

システムアーキテクトと取り組む品質管理

<サブタイトル>

<発表者>

氏名（ふりがな）：福良 智明(ふくら ともあき)
所属： 株式会社オープンストリーム システムインテグレーション事業部 戦略技術推進本部

<共同執筆者>

氏名（ふりがな）： 飯田 秀樹(いいだ ひでき)
所属： 株式会社オープンストリーム システムインテグレーション事業部 戦略技術推進本部 副本部長

氏名（ふりがな）： 大野 泰代(おおの やすよ)
所属： 株式会社オープンストリーム システムインテグレーション事業部 戦略技術推進本部

氏名（ふりがな）： 高島 奈美(たかしま なみ)
所属： 株式会社オープンストリーム システムインテグレーション事業部 戦略技術推進本部

<主張したい点>

システムアークキテクトと品質管理者が連携可能な組織にした結果、テスト工程の効率化と成果物の品質が向上した

<キーワード>

テスト効率化、品質管理者、システムアーキテクト、テスト観点の補完、テスタビリティの向上

<想定する聴衆>

品質保証の担当者、システムアーキテクト

＜活動時期＞

2015 年 7 月～継続中

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

■ 着想の段階（アイデア・構想の発表）

□ 改善活動を実施したが、結果はまだ明確ではない段階

■ 改善活動の結果が明確になっている段階

□ その他（ ）

<発表内容>

- 概要

弊社では、最初数名のテストエンジニア(品質管理者)だけでテスト・品質に関する活動を行なっていました。そこから 2015 年にプロジェクト推進部というテストエンジニアとシステムアーキテクトの一部が連携する組織が立ち上がりました。さらに 2017 年からは、戦略技術推進本部というテストエンジニアとすべてのシステムアーキテクトが配属される組織になりました。現在は、テストエンジニアとシステムアーキテクトが連携しながら、活動を行なっております。
この発表では、これまでテストエンジニアだけで行っていたテスト・品質に関する活動をシステムアーキテクトと連携して行うことによって、テスト工程の効率化と設計書、テスト仕様書の品質が向上した事例について説明します。

- 組織が立ち上がるまでの活動内容と課題

テストエンジニアのみで活動していた時期の活動内容と課題

- 活動内容
  - 社内の受託プロジェクトのテスト工程に横断的に参画し、ノウハウの収集と課題の共有を行う
  - 社内の受託プロジェクトに対して、プロセス、ドキュメントの標準化を進める
  - テスト工程における作業(テスト仕様書の作成と実施　および、テスト工程の管理)を担当する

- 課題
  - 活動の制限
    プロジェクトに参画できるのが、テスト工程の直前である為、ミッションがいかにテスト工程をやりきるかになってしまい、本来やりたかった改善提案が行えなかった
  - 手戻りの発生
    テストベース等テストに必要な情報が直前にならないと入手できないので、テスト実施に必要な情報がない、テストに必要な環境が揃っていないなど手戻り作業が発生し、スケジュールがいきなり遅延しまう
  - テストベースの品質低下
    PM/PL を含めたプロジェクトメンバが開発に関する作業に専念せざるを得ない状況が多く、設計書の品質がテストベースとして利用する為のレベルになっていない
  - テスト観点の重複
    開発者テストがどんな観点でどこまで確認を行っているかの情報が分からず、あと工程のテストと観点が重複してしまう
  - 作業の効率化の推進
    非プログラマが多いので、ちょっとしたツールを作成することによる作業の効率化ができない
  - 開発技術にあったテスト
    技術に関する知識不足により、新技術に対してのテストや開発スタイルにあったテストが行えない

- 課題に対しての取り組み内容と効果

組織が立ち上がり、テストエンジニアと連携可能になってからの活動内容と効果

  - プロジェクトの情報共有
    組織のミーティングの場等で、プロジェクトの状況に関する情報を行った
    → テストエンジニアも参画前に状況やプロジェクトが抱える品質的な問題について分かるようになったので、 事前に課題に対しての準備が行えるようになった(テスト環境の準備依頼、参画後に必要な作業の洗い出し、また成

果物のレビューへの参加など)

★活動の制限、手戻りの発生に対して効果あり

- システムに関する情報の共有

  テスト対象のシステムに関する設計方針、アーキテクトについて共有おこなった

  →開発者テストの方針が分かり、観点の重複を防ぐことができた。また、双方から必要な観点を補完することができ、効率化につながった

  また、開発、テストが協力しながら設計を行うことにより、設計書の品質向上につながった

  ★テストベースの品質低下、テスト観点の重複に対して効果あり

- テストエンジニアの参画時期の前倒し

  品質に関する要求が高いプロジェクトに関しては、テストエンジニアをテスト実施よりも前の工程から参画できるようになった

  ★テストベースの品質低下、テスト観点の重複に対して効果あり

- 技術情報の共有

  テストエンジニアもシステムアーキテクトから最新の技術情報を得ることができた

  ★開発技術にあったテストに対して効果あり

➢ 事例紹介

上記の効果に加えて、テスト観点の補完やテスタビリティの向上にも効果があった事例を紹介

➢ 課題と今後の予定

- テストエンジニアのプロジェクト初期からの参画を常態化させ、テストベースとテストの品質を向上させる
  （テストエンジニアのレビュー参加、テスト分析・テスト設計を行う）
- 品質に関するリスクを早期に発見できるようなプロセスと仕組みを構築する
- 冗長な作業や定常作業の自動化を進め、作業品質の向上と効率化を進める
- 品質を評価する為の指標の定義とメトリクスを効率的に収集できる仕組みづくりを行う

3C2「品質目標と工程別活動内容シートの活用事例の紹介」相澤武（インテック）

<タイトル>

品質目標と工程別活動内容シートの活用事例の紹介

<サブタイトル>

なし

<発表者>

氏名（ふりがな）：相澤　武（あいざわ　たけし）

所属：株式会社インテック　生産本部　品質保証部

<共同執筆者>

なし

<要旨・主張したい点>

本発表は、SPIJapan2016 で筆者が発表した「品質特性に基づく品質メトリクスの定義と活用」の続編である。今回は、考案した「品質目標と工程別活動内容シート」を実プロジェクトに適用してわかった課題とその対策、対策から導き出したプロジェクトへ適用する際の工夫点を活用事例として紹介する。

<キーワード>

SQuaRE、ISO/IEC 25010、品質特性、品質メトリクス、品質計画書、品質目標

<想定する聴衆>

品質保証の担当者

＜活動時期＞

2015 年 3 月～継続中

<活動状況>

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

<発表内容>

1.背景

近年の IT 業界の動向をみると、IT を活用した製品やサービスの種類や社会における役割が増えるにつれ、利用者が求める品質も、従来の必要な機能が正しく動作すること、といったものから、機能の正確性だけにとどまらず、利用目的や利用場面に応じて、安全性やセキュリティはもとより、快適さや楽しさ、また、ビジネスへの高度の貢献といったように多様化してきており、品質を取り巻く環境が大きく変化してきている。

これらの環境変化に対応すべく、新たな品質に対する考え方として、品質特性を理解したうえ、お客さまとの共通の評価基準を持ち要求達成に向け、開発工程の中でどのように品質を担保していけばよいかを計画し、実行するための手法を考案した。（参考文献 1 参照）

考案した手法のキーとなる 2 つのワークシートの特徴は以下の通りである。

（1）要求事項一覧シート

「要求事項一覧」とは、お客様の業務や構築するシステムやサービスの特性に応じて求められるひとつひとつの要求事項を、品質特性・品質副特性に対応付けしたものである。特徴として次の 2 点がある。

①要求事項を一覧形式で俯瞰できる

要求事項と品質特性・品質副特性との対応付けを一覧形式で俯瞰できることにより、品質特性の観点から抜け漏れがないかを確認することができる。

②「要求事項一覧」と「品質目標と工程別活動内容」の相互参照が可能

「要求事項一覧」の要求事項欄と後述する「品質目標と工程別活動内容」のテーラリング理由欄で、相互に参照するができる。これにより「要求事項一覧」に挙がった要求事項を、確実に「品質目標と工程別活動内容」に展開することができる。

（2）品質目標と工程別活動内容シート

「品質目標と工程別活動内容」とは、プロジェクトの品質目標及び目標達成に向けた工程毎の活動内容を定義したものである。定義にあたっては、社内事例の調査で収集した情報を参考にした。特徴として次の 3 点がある。

①品質要求を品質特性で整理・仕様化

品質要求を品質特性・品質副特性の観点で整理することができる。

②品質メトリクスを用いた評価基準の設定

品質メトリクスの基準値は、客観的に判断できるように数値を原則としているが、数値化が難しいものは、できるだけ主観的な判断が排除できる項目を設定し、「YES/NO」で判定できるようにしている。

品質メトリクスには、ISO/IEC 25010（SQuaRE）で規定されている品質特性・副特性に基づき、社内で利用頻度が高く、かつ汎用的に利用が可能な品質メトリクスを選定した。

③開発の主要なマイルストンでの工程評価や出荷判定の判断に利用

品質目標達成に向けて、工程毎ではどのような活動を行えばよいかを計画時に決めておくことで、各工程の移行判定などで、確認すべき項目として利用することができる。

2.改善したいこと

考案した手法のキーとなる 2 つのワークシート「要求事項一覧シート」と「品質目標と工程別活動内容シート」を実際のプロジェクトに適用するにあたっては、業務特性に合わせた特性別パターン（全社共通の汎用パターンとして準備した 2 つのワークシートを業務特性に合わせてテーラリングしたもの）を事前に準備しておく前提であった。

しかし、社内の多くの部門では特性別パターンの準備ができず、全社共通の汎用パターンをそのまま利用しているのが実情で、そのために以下にあげる課題が顕在化しており、考案した手法をプロジェクトで確実に利用するためには、これらの課題への対応が必要であった。

（1）要求事項一覧シートの作成

「要求事項一覧シート」では、お客様の業務や構築するシステムやサービスの特性に応じて求められるひとつひとつの要求事項を品質特性・品質副特性に対応付けることが必要となる。しかし、品質特性・品質副特性の理解不足により要求事項をどの特性に対応付けてよいか悩むことが多く、うまく対応付けができていない。

（2）50 個の品質メトリクスからのテーラリング

全社共通の汎用パターンとして 50 個の品質メトリクスを定義しているが、50 個という数の多さから、多くのプロジェクトでテーラリングがしきれない、全社汎用として準備したパターンが、システム開発寄りの内容であるため、インフラ基盤構築系のプロジェクトでうまく利用できない、といったテーラリングの課題があった。

（3）品質メトリクスの定義

メトリクスのうち信頼性・成熟性のメトリクスとして分類している、検出バグ密度、レビュー指摘密度、テストケース密度などは、これまでも多くのプロジェクトで利用されている反面、各密度の算出方法はまちまちであった。そのため、プロジェクトでこれまで使用していた算出方法が全社の定義と異なる場合、全社統一の定義の利用に抵抗感があった。

3.改善策を導き出した経緯

課題への対策を検討するにあたって、社内で先行して上記 2 つのシートを実プロジェクトで利用している部門の事例を調査し、どのような工夫をしているかを確認した。

４.改善策の内容

実プロジェクトで利用している部門では、それぞれ部門の業務特性に合わせた特性別パターンを定義しており、上述した課題に対しては、以下にあげるような工夫をしていた。この工夫点を、利用がうまくいっていない部門への改善策として適用できると考えた。

4-1.要求事項一覧シートの作成

「要求事項一覧シート」の作成手順を変更し、品質特性・品質副特性と要求事項を対応付けるのではなく、「品質目標と工程別活動内容シート」で品質メトリクスと要求事項を対応付けるようにした。その後に、「要求事項一覧シート」で要求事項に抜け漏れがないか網羅性の確認を行う手順とした。

4-2.50 個の品質メトリクスからのテーラリング

先行部門では、部門内のプロジェクトの事例を元に、利用頻度の高い品質メトリクスに絞り特性別パターンを整備していた。利用頻度の高いものとしては以下にあげるものがあった。

・信頼性・成熟性のメトリクス

（検出バグ密度、レビュー指摘密度、レビュー工数密度、テストケース密度、テスト工数密度）

・保守性のメトリクス

（ライブラリ管理、保守ドキュメント充足、保守テスト環境充足）

4-3.品質メトリクスの定義

信頼性・成熟性に分類している、検出バグ密度、レビュー指摘密度、テストケース密度などの算出方法は、これまで部門で使用している定義を優先した。結果として、全社汎用パターンで定義している算出式と異なるメトリクスも出てきたが、こられについては、メトリクスの名称を変えることで、全社汎用パターンで定義しているメトリクスと重複しないようにしている。

4-4.「品質目標と工程別活動内容シート」の充実化

当初想定した課題への改善策ではないが、今回先行部門の事例を確認している中で、「品質目標と工程別活動内容シート」の中に、他プロジェクトでも活用可能な、様々なノウハウ（例えば、工程別活動内容で使用しているチェックリストやツールなど）が含まれていることがわかった。

5.改善策の実現方法

改善策を全社に展開するために、先行部門の活用事例をガイドとしてまとめるとともに、その内容を e ラーニング教材として公開する。ガイドの活用方法としては以下を想定している。

・要求事項と品質特性・品質副特性とのマッピング例の参照

・品質目標に対する工程別活動内容の事例（チェックリストや利用しているツールも含む）の参照

6.改善による変化や効果

先行して上記 2 つのシートを利用している部門では、現時点では以下にあげるような変化が出ている。これらの変化は、今後活用事例のガイド化と全社公開により、全社にも波及できるものであると考える。

6-1.工程移行判定、出荷判定での基準としての活用

品質計画書に新たに作成した「品質目標と工程別活動内容シート」を作成することで、工程移行や出荷判定時の評価基準が明確になり、実際に工程移行判定や出荷判定の際の評価基準として利用している事例があった。

6-2.品質メトリクスの充実化

全社共通の汎用パターンには無い新たな品質メトリクスを定義する動きが出てきた。これまでは、部門毎に個別にメトリクスを定義していたとしても、全社共有されることはなかったが、「品質目標と工程別活動内容シート」に情報を集約していくことで、全社で共有できる品質メトリクスの事例の充実が可能となる。

部門独自追加したメトリクスとしては以下にあげるものがあった。

・信頼性・成熟性のメトリクス

バグの偏り:単体バグの発生が特定の作成者に偏っていた場合、その原因の究明と対策が採られていること

見逃しバグ:前工程で対処すべきバグが見つかった場合、その原因の究明と対策が採られていること

7.改善活動の妥当性確認

今回紹介した内容は、まだ一部の部門での事例ではあるが、適用したプロジェクトでは、

・お客さまからの要求事項を品質特性の観点で整理する

・品質目標は立案するだけでなく、目標達成のための活動内容も合わせて考えることで、目標達成に向けた活動を確実に行う

などの点を意識してプロジェクトの遂行ができるようになり、これを継続していくことで、多様化するお客さまの要求事項にも応えられるのではないかと考える。

今後の課題としては、適用したプロジェクトでの適用結果の評価や事業部門毎の業務特性に応じたパターンの整備等が必要であると認識している。

A.参考文献

1.相澤武、品質特性に基づく品質メトリクスの定義と活用、SPIJapan2016、2016.10
http://www.jaspic.org/event/2016/SPIJapan/session3B/3B3_ID007.pdf

2.独立行政法人情報処理推進機構（IPA）技術本部 ソフトウェア高信頼化センター（SEC）、
つながる世界のソフトウェア品質ガイド、独立行政法人情報処理推進機構（IPA）、2015.05

3.SQuBOK 策定部会（編集）、ソフトウェア品質知識体系ガイド -SQuBOK Guide-（第 2 版）、2014.11

4.経済産業省ソフトウェアメトリクス高度化プロジェクトプロダクト品質メトリクス WG、
システム／ソフトウェア製品の品質要求定義と品質評価のためのメトリクスに関する調査報告書、2011.03

以上

3C3「「なぜ」に頼らない「なぜなぜ分析」のすすめ」渡辺聡美（富士通エフ・アイ・ピー）

<タイトル>

「なぜ」に頼らない「なぜなぜ分析」のすすめ

<サブタイトル>

～レジリエンスを高め、より高い再発防止効果を生むために～

<発表者>

氏名（ふりがな）：渡辺　聡美（わたなべ　さとみ）
所属：富士通エフ・アイ・ピー株式会社　サービスビジネス本部　センターサービス統括部
明石 DC オペレーション部

<共同執筆者>

無し

<主張したい点>

ヒューマンエラー防止のため、ＩＴサービスに関する各種標準の導入、医療・航空・交通業界に学ぶ人間重視のプロセス整備（仕組み作り）を進めてきた。さらには、意欲向上を促進する取り組みを並行することで品質は大きく改善し、ヒューマンエラーは撲滅された。

しかし、少し広い範囲に目を向けると、全ての組織でヒューマンエラーが撲滅されたわけではなく、品質が大幅に改善した組織においても、ヒューマンエラー発生時の現場要員の分析力低下という悩みが依然残っていることを確認した。

本発表では、当社運用部門の全体品質向上のため、組織横断で取組んだ分析力低下への効果的なアプローチ事例として「なぜ」に頼らない「なぜなぜ分析」を紹介する。その手法は、従来のなぜを繰り返す手法を決して否定するものではなく、高レベルの分析を誰もが実施できるレベルに引き上げるための指導方法として提唱するものである。

さらには、その実践を支えるための場として「教育」のあり方を見つめなおし、熟慮的行動を促進するための双方向教育とするためにどのようにデザインすべきか、一例を紹介する。

<キーワード>

ヒューマンエラー防止、意欲減少、意欲向上、動機付け、なぜなぜ分析、質問技法、監査技法、双方向教育、訓練、トレーニング、レジリエンス

<想定する聴衆>

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

＜活動時期＞

2015 年度下期　現状分析～解決策立案～教育プログラム開発
2016 年度　　部門内教育実施、グループ内への教育プログラム提供

<活動状況>：発表内容に複数の事例が含まれる場合は複数選択可能です。

□ 着想の段階（アイデア・構想の発表）
□ 改善活動を実施したが、結果はまだ明確ではない段階

■ 改善活動の結果が明確になっている段階
□ その他（　　）

<発表内容>

1.背景

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

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

それでもヒューマンエラーは撲滅できず、その原因を考察する中で、要員の意欲減少が影響していることを確認した。そこで、「意欲向上が品質向上の決め手となる」という仮説を立て、意欲向上を目的とした活動を展開。その結果、悲願であった「ヒューマンエラー撲滅」を達成することができた。

しかし、少し広い範囲に目を向けると、全ての組織でヒューマンエラーが撲滅されたわけではなく、品質が大幅に改善した組織においても、ヒューマンエラー発生時の現場要員の分析力低下という悩みが依然残っていることを確認した。その悩みはヒューマンエラーが撲滅された組織にとっても潜在する問題であると推察され、その悩みの解消に向けた取り組みが急務となった。

2.改善したいこと

ヒューマンエラーが減少している組織が抱える「分析力低下」という問題を深堀する中で、以下のような問題が検出された。

・エラー件数が減少することで、実戦的な経験が不足している。
・要員交替もあり、実践的に分析を経験したことの無い者も増えている。
・実戦経験が積めない状況を補う場としての教育が実施されていない。
・「なぜなぜ分析」を義務付けているものの「実践」を指導出来る者が少ない。

一方でヒューマンエラーが発生した際「分析」を行う現場要員の状況を確認したところ、品質マインド低下という実態も垣間見えてきた。

・「なぜなぜ分析」のやり方は知っているつもりだが、うまくいかない。
（なぜのツリーが広がらない。当事者の記憶が曖昧で手戻りが多い。）
・再発防止策による業務負荷の増大。
・ヒューマンエラーを起因としたバーンアウト予備軍

これらの問題を解決することで、効果的な再発防止に繋がる分析力向上という効果をもたらすことが期待された。

3.改善策を導き出した経緯

分析力低下の低下という問題の解決策として一般的にあげられるものは教育開催だろう。しかし、解決策の効果を一層高いものとするために、現状についての「なぜなぜ分析」を行った。それにより、分析力低下の裏に品質マインド低下という問題が潜在し、教育開催のみでは解決できず、「なぜなぜ分析」の仕組みの見直しまでが必要という考えに至った。

また、今回の改善策の中心となる「なぜ」に頼らない「なぜなぜ分析」の仕組み作りに考えが至ったもう一つの経緯は些細なことにある。恥ずかしながら業務ではない。プライベートにおける自身の失敗体験の振り返りだ。失敗に対する周囲の者からの“叱責”、“問い詰め”。「なぜ、●●しなかったのか？」「なぜ、●●したのか？」。表面上は問いかけの言葉だが、自分自身も悔やんでいるところに投げかけられると、その言葉は責めの言葉として心に刺さり、どんどん塞ぎ込んでいった。悲しみがつのり、注意力も散漫になり、行動しようとする意欲は減退し、さらに自分自身を責め続けた。いわばバーンアウトに陥った状態であった。そのようなつらい経験があったにもかかわらず、家族の失敗に対し、自身が「なぜ？」と詰問してしまっていたことにたまたま気付くような出来事が起きた。プライベートでのこの偶発的な出来事により、組織で実施されている「なぜなぜ分析」が現場に同じような悪影響を及ぼし

ているのではないかという疑問を持つに至った。

そこで、組織で実施されている「なぜなぜ分析」について「なぜ」の弊害を中心に確認を行った。なぜなぜ分析は、再発防止の効果的な手法として全社的に推進していた。しかし、多面的な分析の必要性を「知識」として伝えてはいるものの、その実践について十分な指導ができていないという現実に加え、ヒューマンエラーの要因分析に際し、当事者に対する「なぜ」がしつこいまでに繰り返されていることに気付いた。「なぜ」の繰り返しに弊害が無いか、色々な立場の方達にインタビューを行った。すると、ソフトウェア障害のような事象についての「なぜ」は適切に作用するが、ヒューマンエラーについての「なぜ」の繰り返しは、人の意欲を減少させ、要因分析を阻害しがちであるということが見えてきた。そこで、教育に先駆け、「なぜ」に頼らないなぜなぜ分析の確立を目指すこととした。

4.改善策の内容

組織では発生した問題の要因分析の手法として「４M要因分析」、「なぜなぜ分析」、「４Step/M による対策策定」を採用していた。それらの仕組みは十分な期間適用され、その効果が認められ、継続適用することが決定していた。

そこで多面的な分析手法である「４M 要因分析」、戦略的な再発防止策を策定する手法である「４Step/M による対策策定」を効果的に実践するため、「なぜ」に頼らない「なぜなぜ分析」を誰もが実施できる状態を目指し、以下の取組みを進めることとした。

・「なぜ」に頼らない「なぜなぜ分析」の仕組み作り

・「なぜ」に頼らない「なぜなぜ分析」の実践に繋げるための教育提供

１）「なぜ」に頼らない「なぜなぜ分析」の仕組み作り

習熟度や経験によらず、均質な分析ができるような仕組みとするために、分析が得意な人と分析が苦手な人の分析現場を観察し比較した。分析の質に最も大きな影響をもたらしていると考えられた違いは「事実確認」と「原因分析」の工程が独立しているか否かと事実確認の中で利用される「なぜ」の頻度であった。分析が得意な人は、事実確認工程では「なぜ」という言葉をほとんど使わずに事実確認を進め、原因分析から事実確認への手戻りはほぼ発生していなかった。一方、分析が苦手な人は、「なぜ」を多用し、事実確認と原因分析が区別なく実施され、不明点が出ると事実確認に戻るということを繰り返していた。

現場要員へのインタビュー結果とこの比較結果からも、「なぜ」に頼らない「なぜなぜ分析」の仕組みの確立が要因分析の成否を分けると考えられた。

２）「なぜ」に頼らない「なぜなぜ分析」の実践に繋げるための双方向教育の提供

知識を机上の知識で終わらせず、業務において実践するためには、再構築された仕組みの教育を効果的に展開することが求められる。その為にも「周知」という言葉に代表される一方通行の知識の伝達ではなく、習得した知識を実践できる素養を身につけることができる双方向の教育の場とすることが重要課題であると認識した。

5.改善策の実現方法

１）「なぜ」に頼らない「なぜなぜ分析」の仕組み作り

まずは、要因分析という視点は忘れ、ヒューマンエラー発生に至った経緯について事実確認を行う仕組みを確立する。従来のヒューマンエラー発生の際の事実確認の中心は当事者ヒアリング（聞き取り調査）であった。しかし、事実が洗い出されないばかりか、当事者のモチベーションまでもが低下してしまうというような失敗ケースもしばしば発生していた。そこで、従来のヒアリングに依存した事実確認について抜本的見直しを行うこととした。参考としたものは「質問技法」と「監査技法」である。これらは、ビジネスシーンでは多く活用される技法であるもののヒューマンエラー対応のシーンではあまり活用されていない。しかし、これらの技法を意識して活用することで、事実確認の精度を上がることが期待された。

○質問技法の活用

当事者へのヒアリングにおいては、「尋問を受けている」という印象を与えないようにすることが事実確認の精度を上げるための第一の留意点となる。そのための基本スタンスは「傾聴」。適宜オープンクエスチョンとクローズドクエスチョンを組み合わせながら、当事者の記憶を出来る限り正確に辿れるように支援するという役割がインタビュアーには求められる。オープンクエスチョンにおいては「なぜ」を極力除いた「４Ｗ１Ｈ」を活用し、クローズドクエスチョンで締めくくることで、当事者とインタビュアーの認識が一致していることが確認できる。また、ヒューマンエラーを取り巻く周辺要素を網羅的に確認するために４Ｍ（人、物、環境、管理）をオープンクエスチョンのシーンで活用することが効果的であると考えられた。

○監査技法の活用

当事者へのインタビューで得られる情報は、全てが事実であるとは限らない。それは、恣意的な情報操作という意味ではなく、記憶錯誤が含まれるということを意図している。収集する情報の精度を高めるためには、監査技法を組み合わせることが期待される。インタビューで得られた情報が有効な情報であるか、記憶錯誤が含まれるかを「閲覧（文書や記録）」、「観察（日ごろの振る舞い）」、「再実施」と組み合わせながら評価することで、有効な情報が選別されると考えられた。

このように事実確認を進めることで、ヒューマンエラーの背後要因が多数浮かび上がった状態となる。これらの背後要因に対して対策を講じても一定の効果は得られるが、より効果を高めるためには、これらの背後要因を論理的に分析した後、根本原因に対する戦略的な対策を講ずることが早道となる。言い換えると、事実確認で集めた様々な事実を「なぜ」そうなったのかと問いかけながら整理するのである。いわゆる「なぜなぜ分析」のアウトプットは、当事者に「なぜ」と問いかけて作成するものではなく、集められた事実を「なぜ」で整理することで作成されるという考え方だ。このように意識を変えることにより、「なぜなぜ分析」の難易度は下がり、比較的容易に根本原因にたどり着くことができるようになった。

２）「なぜ」に頼らない「なぜなぜ分析」の実践に繋げるための双方向教育の提供

組織で採用している「なぜなぜ分析」を効果的に現場で実践していくためには知識の伝達のみでは不十分である。知識をスキルとして定着させるためには、教育の場とは別に継続した訓練（トレーニング）を行うことが必要となる。このトレーニングを自発的に継続する習慣をつけることで、分析力が向上し、業務で実践する素養が身につく。そこで、教育の場をその動機付けの場となるようデザインすることとした。キーワードは振り返りとコミュニケーション。講師が受講者に対して一方的に教え込むのではなく、講師と受講者の間、あるいは受講者同士の積極的なコミュニケーションが存在する双方向教育とするために演習のデザインを模索した。

＜演習例＞

テーマ：自身の成功体験は失敗に対する一連のプロセスがもたらした成果であることを気付かせる

構成：振り返り：繰り返してしまう失敗と繰り返していない失敗を思い出し、対比する

コミュニケーション：繰り返したくない失敗に対する発生後の対応に対する相互アドバイス

失敗しない人はいないので、誰もが発生後の対応をトレーニングできる場を持っていると考えられる。しかし、全ての失敗に対して主体的な対応を求めると息が詰まってしまう。そこで、自身が繰り返したくないと考える失敗を起こした際に、業務プロセスを活用することを推奨した。そうすることで、当事者にとっては繰り返したくない失敗が撲滅でき、かつ、業務においていつでも実践できる素養が身につくという一石二鳥の効果がもたらされた。

6.改善による変化や効果

前述の改善は品質状況の異なる様々な組織にそれぞれの効果をもたらした。

１）　障害多発という悩みを抱える組織

・失敗に対して意味づけを行い、次に生かそうとする熟慮的行動の促進

→失敗で一時的に落ち込んだ状態からの回復（レジリエンス）の促進

・現場要員に根付いていた「やらされ感」の排除
→バーンアウトの回避
・失敗してチーム全体が落ち込んだ状態になっても、一致団結して乗り越えようとする意識の促進

2） 障害が減少する中で分析力低下という悩みを抱える組織
・失敗に対して意味づけを行い、次に生かそうとする熟慮的行動の促進
→失敗で一時的に落ち込んだ状態からの回復（レジリエンス）の促進
・分析力向上（差し戻しの減少）
→障害対応のスピードアップと工数削減

3） 安定した品質を維持している組織
・経験不足による分析力低下の抑止
・未然防止活動における熟慮的行動の促進
→失敗ゼロ継続期間の伸長

＜熟慮的行動の一例＞
・分析力向上による再発防止と類似障害未然防止の促進
・探究心向上による改善促進

7.改善活動の妥当性確認

今回の改善活動を通して、より高い品質を実現する活動の原点は「なぜなぜ分析」にあり、その効果的な実践を促進するために"意欲"をコントロールすることが重要課題であるという気付きを得た。

冒頭にも述べたように、データセンターの重要性は年々高まっている。本発表では、失敗を生かし、成功している組織からの学びにより品質を向上させるという点について述べてきた。残る不安要素は意図的な行為（不祥事）に起因する品質低下である。

一見、方向性の異なるテーマのようにも見えるが、実は不祥事の防止にも"意欲"が影響していることが各界で延べられており、今回の活動は当社の背景に対し、妥当な改善策であったと評価できる。

失敗、不祥事何れにしても、問題が発生してしまった場合の損失は測り知れないほど巨額なものとなり得る。仕組みの導入や定着のためには、一定のコストが必要となるが、必要なコストの大半は双方向教育の継続実施のための工数である。教育のためのコストを要しても、その後の様々な改善活動が促進されることを考慮すると、教育のためのコストの費用対効果は高いと推察される。

また、当事例は運用部門に限らず、グループ会社への展開にも着手し始めており、ヒューマンエラーに対する「なぜなぜ分析」の質向上を推進する手法として有効であるという評価を多く受けている。その評価からも、今後より多くの組織への展開が期待される改善活動であると考えている。

A. 参考情報

[1]医療におけるヒューマンエラー 河野龍太郎著 医学書院（2014 年 3 月 1 日）
[2]わかりあえないことから 平田オリザ著 講談社（2012 年 12 月 10 日）
[3]「わかりやすい」表現の技術 藤沢晃治著 講談社（2005 年 2 月 7 日）
[4]フィンランド流「伝える力」が身につく本 北川達夫著 中経出版（2010 年 12 月 6 日）
[5]情報セキュリティ監査手続ガイドラインを利用した監査手続き策定の手引 経済産業省（平成 21 年 7 月）
[6]双方向授業は日本の未来を築く（豊中ロータリークラブ主催教育フォーラム報告書）（2011 年 1 月 22 日）