

クオリティ ワン
Quality One

Vol.11 2010年8月号

Software Quality Profession

財団法人 日本科学技術連盟

1. 品質

■ 混乱からの脱出 ～ XDDP で現場は甦った ～

(株)デンソー技研センター

古畑 慶次

はじめに

前号で清水吉男氏より XDDP の理論が紹介されました。今月号では、(株)デンソーにおけるナビゲーション開発への XDDP の導入事例を紹介します。

XDDP の導入を開始して、今年で 4 年目となります。効果検証の試行プロジェクトから始まり、希望者へのコンサルティング、そして、リーダー育成と一つのコンセプトに従って展開をしてきました。4 年という時間はかかっていますが、正しく導入したプロジェクトでは、品質、生産性について確実に成果が出ています。そして、何より大きな変化は、何人かの技術者については、開発への取り組みが確実に“前向きになった”という嬉しい事実です。

開発で多忙の中、社外発表や論文作成に技術者が自発的に取り組み始めました。今年は、XDDP 関連だけで社外発表は 3 件、論文作成は 2 件を予定しています。また、コンサルティングに使用する成果物の内容や送られてくるメールの送信時間を見ると、自ら努力を始めた技術者の姿を容易に想像できます。

まだ、“変化の兆し”というレベルですが、こうした技術者が組織に 16% 出て来れば（後述）、現場は甦り、技術者が本来の仕事を取り戻せると考えています。タイトルは、敢えて“現場は甦った”としました。彼らへの敬意と確信とを込めて、これまでの取り組みを紹介したいと思います。

混乱する開発現場

XDDP を導入した開発現場は、CMMI のレベル 2 を達成した組織であり、プロセスは定義され、開発用のドキュメント類は多数作られていました。しかし、結合テストやシステムテストで大量のバグが発見されるのです。これらのバグを修正するには莫大な時間が必要でした。技術者たちは、自分の時間の多くをバグ修正に費やすことで、品質の高いシステムを作り上げていたのです。

バグが収束しなければ、当然、次の開発に影響が出ます。前機種のバグ対応が終わらないうちに、次のプロジェクトがキックオフされるため、技術者は新しい案件に着手できません。結局、新しいプロジェクトでも要件開発や設計の時間が十分とれず、テスト工程でバグが大量に出てしまいます。このように負のスパイラル(デスマーチ)から抜け出せない状態でした。

こうした開発の中で、技術者は現場を奔走する毎日を送っていたのです。そして、常に高負荷を強いられ、疲弊していました。繰り返される不具合や仕様変更により、新しい技術の習得や現状の分析をする時間を奪われ、自ら何かに取り組む気力を失っているかのように見えました。

この状況を打開するには、テストに入る前に、仕様の抜け漏れ等の不具合を“気づく仕掛け”、あるいは、“気づいていないことを気づかせる仕掛け”が必要でした。そんな中、XDDP と出会ったのです。そ

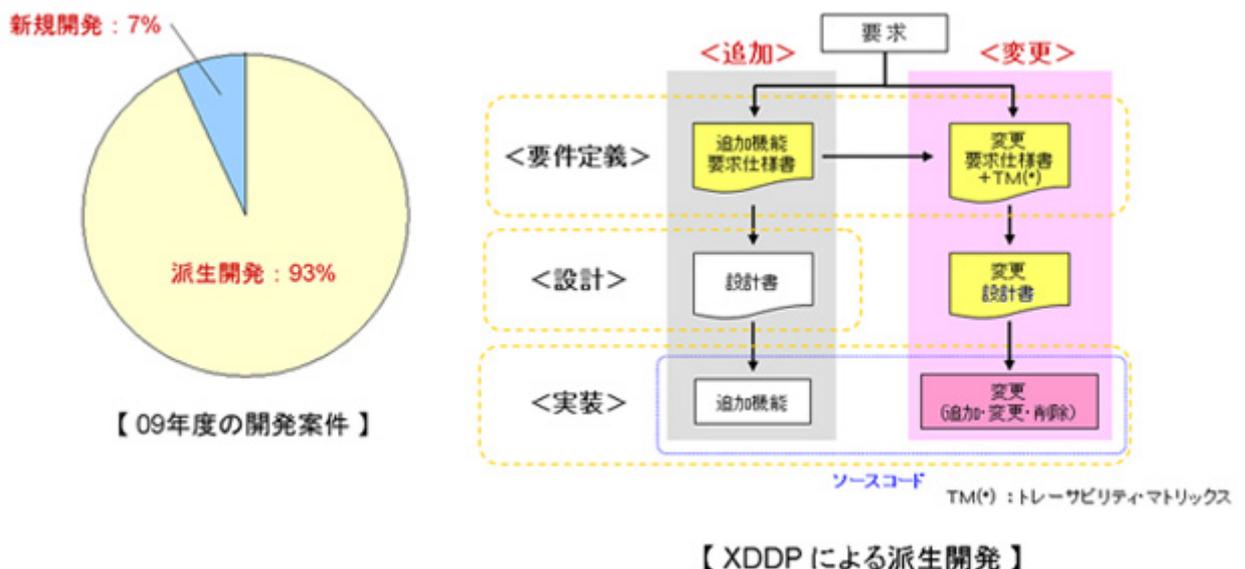
して、“これなら行ける”と直感しました。

今、着目すべきは“派生開発”

XDDP は、派生開発（前号参照）用に提案された開発プロセスです。実際の開発における新規開発と派生開発の割合を調べてみると、ほとんどが派生開発であることがわかりました（図：09年度の開発案件）。しかし、XDDP を展開した組織のすべての枠組み（仕組み、プロセス、成果物等）は、新規開発向けに作られたものでした。確かに、清水吉男氏の本を除いて、ソフトウェアエンジニアリングの書籍は、新規開発を対象に書かれたものばかりです。CMMI のレベル 3 で必要となる標準プロセスを考えてみても、派生開発用の標準プロセスが存在している組織は少ないのではないのでしょうか？

現実的に考えれば、新規開発よりも派生開発を中心に考えていく必要があります。新規開発と派生開発では、顧客からの要求や制約条件が全く異なりますので、派生開発に新規開発のプロセスを適用するには無理があります。派生開発では、新規開発のように仕様書や設計書を文章として完成させておけば、変更点や影響範囲の抽出が難しくなるのです。新規開発のように変更点を考慮して仕様書を作成しても、派生開発では、次の設計のフェーズで、またその変更点を抜き出して設計書に反映させます。こうした作業では、ドキュメントを作ることに注意が払われ、変更点や影響範囲に抜け漏れが出てしまいます。これが、テストでバグが大量に出ていた要因の一つでした。

新規開発では、要求仕様書や各設計書を完成させて開発を進めます。しかし、派生開発で必要なのは差分情報です。新規開発のプロセスを使って、この差分情報を要求仕様書や各設計書に入れ込み、各ドキュメントを完成させながら派生開発を行えば、時間が足りなくなります。その結果、途中のプロセスを飛ばして、ソースコードの変更に取りかかるようなことが起こるのです。



開発現場から見た XDDP

我々は、XDDP の導入を決め、試行プロジェクトにおいて、ある程度の手応えを得ました。実際に現

場に導入してみて、XDDP について感じたのは以下の 3 点です。

(1) プロセスが極めて合理的である

XDDP は、派生開発を、変更と追加というタイプの異なる 2 つのプロセスに分け、それぞれの要求を実現する最適なプロセスと成果物を提供しています (図: XDDP による派生開発)。特に、変更のプロセスでは、変更仕様とその影響範囲の抽出を表記法を工夫したり、レビューをうまく活用することで、設計者の思い込みや勘違いを最小限に抑えます。XDPP は、要求を実現するという観点から、必要なプロセスを層別し、整理した極めて合理的なプロセスと言っていると思います。

(2) これまでの現場の問題に的確に対応している

多くの組織では、テストでのバグや後戻りが頻発しています。XDDP は、これらの問題の根本原因を解消するプロセスと成果物で構成されています。それは XDDP を導入したプロジェクトの品質と生産性から判断できます。我々の試行プロジェクトでは、従来の開発と比べて、テスト工程での不具合発生率 65% 減、生産性 1.3 倍という結果を得ました

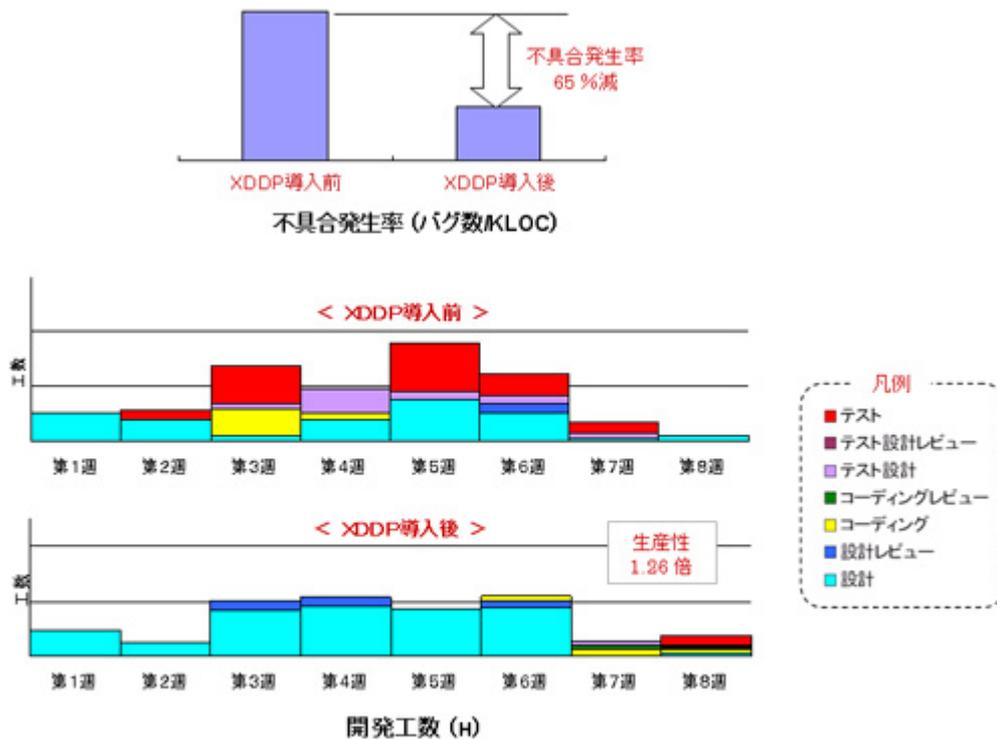
(図: XDDP 導入による効果)。

(3) トヨタ生産方式の考え方にマッチしている

トヨタ系のメーカーとして感じるのは、大野耐一氏が考案したトヨタ生産方式 (以降、TPS: Toyota Production System) の考え方によくマッチしているということです。これは、いくつかのプロジェクトで実際に取り組んでみて気づきました。TPS は、ムダの徹底排除とプロセスの合理性を追求した生産方式ですが、その点では、XDDP も同じ思想で作り上げられた開発プロセスと言えます。

また、TPS では、常にプロセスを変化させること (継続的改善) が求められます。“時代の変化を見据え、お客様の要求にあわせてプロセスを変えていく”、この発想により、TPS は製造現場を“強靱な生産システム”へと変貌させます。

これは、XDDP でも同様です。派生開発では、毎回、要求や納期、担当者が変わるので、前回のプロセスをそのまま使うわけにはいきません。さらに、派生開発では開発期間が短いので、要求を実現する確かなプロセスが必要です。XDDP では、PFD (Process Flow Diagram) を使ってプロセスを設計し、こうした変化に対応します。



【 XDDP導入による効果 】

現場展開の考え方 ～「イノベータ理論」にもとづく改善戦略～

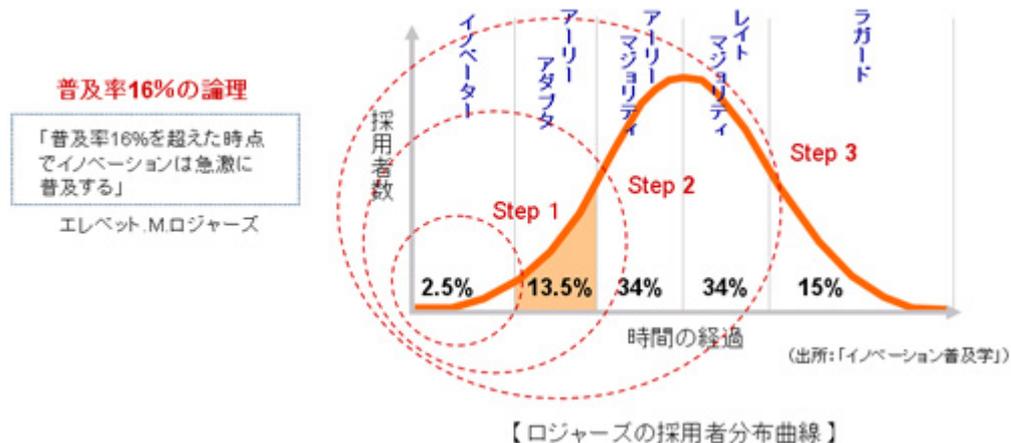
XDDPについては、その技術もプロセスも理解しました。また、試行プロジェクトにより、現場への効果も確信できました。あとは、“開発現場の技術者にどう納得して取り組んでもらうか”が大きな課題でした。

その時、参考にしたのが、エレベット.M.ロジャーズの「イノベータ理論」です。イノベータ理論では、「普及率 16%を超えた時点でイノベーションは急速に普及する」（普及率 16%の論理）としています（図：普及率 16%の論理）。そこで、XDDP を一つのイノベーションと捉え、ロジャーズの採用者分布曲線に従った展開を今回のプロセス改善のコンセプトとしました（図：ロジャーズの採用者分布曲線）。つまり、全体の底上げを狙うのではなく、イノベータから始め、アーリーアダプタを育て、そして、アーリーマジョリティへと XDDP を順次展開していく戦略をとったのです。

全体の底上げによる展開も考えましたが、推進側のリソースが限られているため、現場全体へは十分な対応はできません。しかし、普及率 16%の論理に基づき、最初は、2.5%のイノベータを対象にし、次にアーリーアダプタに切り替えていけば、推進側のリソースが十分ではなくても、XDDP を使いこなせる技術者を育てることは可能です。そして、そうした技術者が先行して育てば、彼らと一緒に展開活動を推進していくことができるのです。現場の技術者を指導者に育て、彼らと共に現場に“教え合う文化”を醸成する。まさに、これは、少人数のスタッフで大組織のプロセス改善に立ち向かう我々にふさわしい方法でした。

そこで、まず、全体の16%の普及率を目指し、次のようなステップで進めました。

- Step 1 : 関心のある人から導入 (イノベータ)
- Step 2 : リーダーを育成 (アーリーアダプタ)
- Step 3 : 草の根活動へ展開 (マジョリティ)



現場の何が問題か? ～ 個人の習慣と組織文化に挑む ～

改善戦略は立案できましたが、問題はこれをどう実践するかです。技術者に対して何をどう考え、具体的にどう行動していくのか、推進側が取り組む姿勢を変えなければ、現場の技術者は本気で取り組んでくれません。

これまで XDDP の現場展開を進めてきて感じるのは、技術的なハードルよりも、むしろ人の意識に問題があるということです。これまでのプロセス改善活動や CMMI や ISO の取り組みで、技術者に改善への拒否反応やトラウマを植え付けてしまっている可能性があります。今回、XDDP の導入前に、現場の技術者にプロセス改善について聞いてみると、以下のような話が聞けました。

- CMMI のレベル 2 をクリアしたが現場はよくなる
- 決められたドキュメントは作るが何に使うかわからない
- どうせまた、しばらくしたら違う方法を進めるのだろう
- 推進側は現場のことを理解していない

これらは、今回の XDDP の展開についてのものではないですが、現場の技術者の推進側に対する不信感をよく表しています。CMMI や ISO 自身は決して悪くないのですが、推進側が現場への導入を誤り、正しく誘導できていない可能性があります。こうなると、むしろ今回の取り組みは推進側への信

頼を取り戻すことを前提に、我々推進側と技術者の考え方や習慣を変えることが成功への近道だと感じました。そして、“技術者の習慣を変え組織文化を変革する”をスローガンに、“個人の習慣”と“組織文化”に挑む活動を始めました。

展開の方針 ～ PCA と コミュニケーション ～

XDDP の展開を進めるのに当たり、2つの方針を掲げました。

1. PCA (Person Centered Approach)による個別性、独自性の尊重
2. コミュニケーションを主体にした信頼関係の確立

プロセス改善の対象である個人の習慣や考え方をを変えるためには、個別に対応していくしかありません。ソフトウェアの開発では、プロジェクト毎に要求や制約条件が異なります。また、技術者についても、性格、技術、キャリア等、当然違ってきます。こうした状況下で、現場に対して十把一絡げに施策を打っても、問題は決して解決しません。

PCA は、カール・ロジャーズが提唱した、一人ひとりの人間の個別性、独自性を尊重し、あらゆるコミュニティとの積極的な対人関係を深めようとする考え方です。XDDP の展開では、PCA に基づいて、開発状況や技術者の個性を尊重したコンサルティングと技術支援を活動の中心にしました。

そして、そのアプローチの中で重視したのが、コミュニケーションによる現場技術者との信頼関係の確立です。“共感”はするが“同情”はしない。“理解しようとする”のではなく“受け入れる”。技術者が直面している状況を察し、伴走者の気持ちで支援を諦めずに続ける。これらは、我々が目指す目標と現場を分析した結果から、自ずと導かれる方針でした。

人を振り向かせる技術 ～ 営業活動から学ぶ ～

取り組みの中で、一番苦労したのは、Step1 から Step2 への移行です。Step1 では、組織の中で XDDP に個別に取り組んでいる人や、興味のある人が対象でしたので大きな問題はありませんでした。しかし、Step2 では、リーダー候補を選定し、XDDP を指導できる人材に育てる必要があります。

そのためには、XDDP を十分理解していないリーダー候補者に、XDDP に実際に取り組んでもらわなければいけません。XDDP に振り向いてもらわないといけないわけです。このような点から、このフェーズの活動は、“営業”と考えることができます。つまり、この活動は、XDDP という商品を顧客である技術者に買って頂く営業活動であるということです。

営業の技術は、営業活動で成果を上げている中堅企業や書籍から学び、そこで得たエッセンス（以下）を実際の XDDP の展開活動に活用してきました。

- (1) 「訪問時間」よりも「訪問回数」

これまで、何度となく技術者の席まで行き、話をしてきました。その時に注意したのが、“話す時間”より“行く回数”を重視する「訪問時間より訪問回数」という発想です。1時間まとめて話をするよりも、15分を4回に分けて話をする方が効果的だということです。

技術者は、日々“忙しい”と感じていますので、長く時間を取られることを嫌います。そこで、何度も技術者の元に足を運びながら、彼らの話に耳を傾け、状況にあわせて XDDP の話を少しずつしていきます。席にいないときは、来たことを名刺に書いてキーボードの上に置いておきました。これも1回の訪問でした。

何度も席まで行って話をするうちに、少しずつ耳を傾けてくれるようになります。そうして、取りかかれる内容を一緒に考えながら、自ら一步踏み出す瞬間を待ちます。かたくなな態度をとり続ける技術者もいましたが、そういう技術者は、“縁がなかった”と諦めました。

(2) 「言い訳」はチャンス

いざ、取り組みの話題になると必ず出てくるのは、出来ない理由（言い訳）です。「取り組む時間がありません」「全部は無理です」「プロジェクトはまだ途中なので・・・」「取り組むテーマがありません」等、いろいろ聞いてきました。こうした人は、言い訳して、結局何も変えません。そして、皮肉なことに彼の言い訳は当たります。そこに気づかなければ、変わることはできないのです。ですから、出来ない理由に全く意味はありません。それに、この先、できる条件が完全に整うことなど決してないのです。

しかし、こうした言い訳の対応策が工夫できれば、彼らは取り組まない理由を失います。技術者によって異なりますが、一通り話を聞いて状況を共有できたら、“出来ない理由”を“出来る理由”にかえるアイデアを一緒に考えます。こうして外堀を辛抱強く埋めていきます。

ただ、言い訳も2種類あります。“前向きな”言い訳と“後ろ向きな”言い訳です。“前向きな”言い訳には、本当に何とかしたくても何ともできない技術者の気持ちが表れています。ここを感じ取り、共感し、受容する姿勢が推進側には必要であると感じています。

(3) 技術的な説明は妥協しない

XDDP を十分理解していない状態では、XDDP の技術そのものに対する誤解が生じています。そこで、技術的な質問が出た時には、的確かつ丁寧に答える必要があります。「なぜ、うまくいくのか?」、「今までと、どこが違うのか?」、「どんな技術が新たに必要か?」、「習得にどのくらいかかるのか?」等、質問はさまざまですが、すべての質問に納得できるよう答えて、技術者の不安や疑問を解消してきたつもりです。この回答には、推進側と XDDP への信頼がかかっていますので、誠意ある姿勢が求められました。

技術者は、「やりたくないから取り組まない」のではなく、「知らないから取り組まない」という場合がほとんどです。技術を理解してもらうまで粘り強く説明すると、能力の高い素直な技術者は興味を持

ってくれました。まだ、頭の中のどこかに“なんとかしたい”“やってみたい”“うまくやりたい”という気持ちが残っていた証拠だと思います。

こうして、現場の技術者を徐々に“せざるを得ない状況”に追い込み、彼らの気持ちの変化を待ちました。そして、取り組むきっかけをつかんだ技術者は一歩踏み出してきますので、そのタイミングで技術支援を開始しました。気持ちが傾いたその瞬間、そのタイミングで彼らが必要としているものを必要なだけ提供したのです。

世間では、“Time is money（時は金なり）”といますが、開発現場の改善では、“Timing is money（タイミングこそ金なり）”の発想が重要でした。このタイミングを把握でき、こうした機会を意図的に作り出せる人が、組織には専任者として必要だと思います。

“変化の兆し” ～ 変わり始めた技術者たち ～

「普及率 16%の論理」では、16%を超えると組織全体に影響すると言われてはいますが、実は確信があったわけではありません。むしろ 16%を目標に、あらゆる手段を考え実施してきたというのが本音のところですよ。おそらく、周りの誰も信じてはいなかったと思います。

しかし、確実に技術者たちは変わり始めました。「はじめに」では、“変化の兆し”として、論文や社外発表への取り組みを紹介しましたが、今では、次のような技術者も目にするようになりました。

- 身近な話題でテーマを作り、要求仕様書の書き方の“けいこ”を始めた中堅技術者
- 「私も課題を持って発表できるような仕事がしたい」と相談に来るリーダー
- “アーキテクチャ再構築”に自主的に取り組み、コンサルティングを希望する若手技術者
- “なんとかしたい”とオフショアでの XDDP の活用を相談に来る上司と部下
- XDDP の勉強会を自ら企画し、課内で部下と一緒に勉強を始めた管理職

まだ、このような技術者は全体の 16%には至ってはいませんが、こうした光景を見ていると個人の習慣から組織文化が徐々に変化している手応えを感じます。組織の中でも、技術者の変化を話題に上げ、嬉しそうに話をする管理者もいます。「もしかすると 16%を超えたら展開が加速して組織が変わるかもしれない・・・」展開前に想定したロジャーズの仮説が、今、現場への大きな期待へと変わり始めました。

まとめ ～ 混乱から時代の頂へ ～

現在の取り組みについては、紙面の都合でこれ以上説明はできませんが、現場では、今、Step2 から Step3 への活動を引き続き展開しています。これまでの経験から言えるのは、プロジェクトの状況や現場環境、技術者の個性を把握し、彼らの能動をいかに引き出すか、つまり、技術者 1 人 1 人の“自律”と“工

夫”が活動の成否の鍵を握るということです。

自分の意志で一步踏み出した技術者たちは、XDDP の技術を着実に習得し、品質、生産性を改善することで自分の時間を確保します。そして、“次の時代への準備”を始めます。アーキテクチャの再構築や分析手法の習得、それに見積もりや進捗管理、スケジューリング等、やっとエンジニアリングや管理技術に取り組むことができるのです。

私が XDDP の技術支援を積極的に進める狙いは、まさにここにあります。それは、疲弊した開発の中でうつ病に倒れ、多くの人に助けて頂きながら執念で手に入れてきた XDDP への確信であり、この技術こそがソフトウェア技術者を救い、再生する一歩だと信じるからです。

混乱を制した先には、誇りと笑顔を取り戻した技術者が、ソフトウェア開発を心から楽しみ、組織の“希望”となることを願ってやみません。ソフトウェア開発に希望を失った技術者も、本来の能力を発揮し、自分らしく胸を張って生きていける可能性がここに 있습니다。時代の頂に何人の技術者を導くことができるか、XDDP によりプロセス改善に新たな展開が始まりました。

プロフィール

古畑 慶次（こばた けいじ）

(株)デンソー技研センター

技術研修所にて、高度技術者育成のための研修構築、講師を務めながら、開発現場への支援・指導を手がけています。XDDP の基盤技術である要求仕様書の書き方やプロセス設計を中心に、ソフトウェア技術やマネジメント、論文作成、発表指導で若手技術者と格闘する毎日です。現場では、産業カウンセラーとしても活動しています。

2. 人材育成

■ ソフトウェア開発における知の整理・体系化と伝承 —JSQC ソフトウェア部会「遺言」プロジェクト—

(社)日本品質管理学会ソフトウェア部会前部会長

兼子 毅

1. ソフトウェア開発における「知」

日本のソフトウェア産業における主たる生産物は、大きく分けると二種類あるといえるだろう。一番目は、特定の企業の、特定の業務を自動化、省力化、効率化するために開発される受注型業務システムである。これらは、紙媒体の帳票が物理的に部署間を移動しながら、様々な情報が付与されたり、承認されたり、モノが動いたりしながら進められている実際の業務を、完全に、あるいは大部分コンピュータのソフトウェアに置き換えるものである。もう一つは、様々な機械の中にあり、メカニズムやエレクトロニクスの制御を行ったり、簡単なユーザ・インタフェースを提供したりする組込みシステムである。一昔前の自動車では、アクセルペダルを踏み込むと「物理的に」キャブレターの燃料噴射量が増加していたが、いまどきの車では、アクセルペダルは単にボリューム操作のための特別なインタフェースに過ぎず、その踏み込み量を数値化した上で、燃料噴射量を制御している。

多くの開発プロジェクトでは、単純に既存の業務をソフトウェアに置き換える、あるいは物理的なメカニズムを CPU とセンサーからなる制御システムに置き換える、というだけではない。一定の投資を行うからには、単純に省力化、効率化、自動化するだけではなく、何らかの付加価値を与えることが多い。つまり、従来の業務では存在しなかった、あるいは、やりたくても現実的とは言えなかった、何らかの機能や業務が付け加えられていることがほとんどである。

このような開発においては、いくつかの独立した能力が開発者（と発注者）に要求される。

- 1) 発注者が自動化・システム化を望んでいる業務・対象の理解と記述
- 2) 発注者が実現を望んでいる「夢」の理解と記述
- 3) 記述された業務や「夢」を実現するシステムの設計
- 4) それらのシステムを実現するためのコストなどの見積もり
- 5) 限定された期間とコストの中で、どこまで業務や「夢」を実現させるか、発注者との交渉
- 6) 合意されたゴールを、定められた期間とコストで実現するための技術とマネジメント

従来ソフトウェア工学、あるいはソフトウェア科学として研究対象となってきたものは、上記の3)、4)、及び6)である。1)や2)も対象となっているが、実世界の本質的非論理性が、その解決を難しくしている。5)は実際のプロジェクト・マネジメントでは極めて重要なポイントだが、ソフトウェア工学などの文脈でまともに議論されていない。

「実世界の本質的非論理性」について、簡単に触れておこう。例えば、獲得した要求仕様の矛盾を示すことは数学的・論理的に可能である。しかしながら、獲得した要求仕様が、現在の顧客業務を正確に反映しているかどうかを数学的・論理的に立証することは、そもそも不可能である。よく使う例なのだが、ある金融機関で「お客様を優遇します」と言っている。お客様は誰か？何を、どの程度優遇するのか？これらを聞きとって紙に書き留めたとしても、本当にそのように実際の業務が行われているかを「論理的に」確かめる手段はない。支店長決裁で「お客様」認定された顧客もいるだろう。いままでの取引履歴から、一定条件を満足しているにもかかわらず「お客様」認定されていない顧客もいるだろう。全て「明文化」したルールに基づいて業務を運用しているところは、実はあまり多くない。全てが「明文化」されているわけでもなく、実際の運用ではルールと相反する決定がなされることもあり得る。

それでも発注者は、ありのままの業務を、そのまま自動化したい（もちろん、システム化するにあたって「業務改善」を行う場合もある）のである。特にエンタープライズ系のシステムの場合、どこまでもこのような「実世界の本質的非論理性」がついてまわる。発注者が自らの業務に対して最適化しようとすればするほど、このような問題はあちらこちらで出てきてしまう。しかも、このような「人が『恣意的に』決めたルール」は、極めて簡単に「変わる」。ビジネスの都合、競争相手との差別化、様々な要因が、これらのルールを簡単に過去のものにしてしまう。お金を取れるかどうかは別にして、これらに付き合っていかなければならない。

一方、組込み型システムの開発の場合、相手はもう少し「モノ」寄りであることが多い。このような開発の場合、実時間という制約が問題となることが多い。例えば、自動車エンジンの燃料噴射制御では、おおよそ0.1秒ごとに燃料噴射量を計算しなければならない。しかも一気筒あたり、である。限られたハードウェア性能のもとで、高機能化によって設置された様々な種類のセンサーからの情報を集めて、である。

ソフトウェア開発には、このように、いくつかの「難しさ」がある。現実には、日々進められているソフトウェア開発を的確に行うためには、様々な「知」が必要である。いろいろな教科書もあるが、ほんとうに困ったとき、いざという時に役に立つ「知」は体系化されていない。したがって、教育や伝承も極めて困難である。JSQC ソフトウェア部会では、それらの「知」を集め、体系化を試みるプロジェクトを2年ほど前から始めた。その活動の中で私が得た「気づき」をお話したい。

2. フレームワーク（トップダウン型知）と実践（ボトムアップ型知）

欧米型の「知」の多くは、「フレームワーク」の形をとる。全体像がひと目でわかり、構造も明確である。一方日本型の「知」の多くは「実践」の形をとる。一つ一つは極めて具体的だが、全体の位置づけ、構造などは不明確で、したがって、短い期間に他者に伝達することが極めて困難である。家元制度、一子相伝、免許皆伝など、長い期間一緒にいる中で伝えられていく「極意」のようなもの、これが日本型の「知」である。なぜ日本人は「枠組み」を考えるのが不得手なのか。これは本稿の主題から離れるので、あえて触れない。筆者個人的には、麦を主食とせず米を主食としたこと、あたりが関係しているのではないかと考えているのだが、それはまた別の機会にでもお話したい。

さて、ここ20年ほど、日本における品質管理やマネジメントの分野でも、様々な欧米型フレームワークが導入されてきている。それは自分たちの活動を再構成し、整理する上では極めて有益ではあるが、それらのフレームワークが持つ本質的な陥穽に気付いていないため正しく用いられていないことが多いように思われる。フレームワークが持つ本質的な陥穽とはなにか？いろいろな業務や業種で使える枠組みとするために、途中のレベルから下の活動を「抽象化」している、ということである。特定の業務や業種で当該フレームワークを有効に用いるためには、まさにこの抽象化された活動を具体的に記述し、かつ実践することが極めて重要である。しかしながら、ISO9000 シリーズの第三者認証の仕組みなどの影響もあり、この「具体化すべき部分」を放置したままフレームワークを適用するという極めて憂慮すべき状況があらこちらで見られる。

分かりやすさも、欧米型フレームワークの特徴であろう。日本における師弟関係では、きっとこんな会話がなされている。「珍念（弟子の名前です）、まずはこれができるようになりなさい」「できました」「まだまだダメだ。次はこれに挑戦しなさい」というようなやりとりが延々と続き、ある時、お師匠様に呼ばれて「免許皆伝」を告げられる。いつ終わりが見えるのか、わからないまま走り続けなければならない。欧米型だと、もっとスマートだ。「この仕事には、ポイントが3つあります。一番目はこれ、二番目はそれ、そして三番目はあれです。ひとつずつマスターしていきましょう」というような感じだろう。極めて分かりやすい。

誰かが、あなたの仕事のために、専用の枠組みを考えてくれたのなら、それは素晴らしいことだ。考え抜かれていて、必要なことが全て収まっているのだろう。でも待つて欲しい。貴方と全く同じ仕事をしている人は、世界に何人いるのだろうか？その人達のための専用枠組みを、誰かが作ってくれるのだろうか？欧米型フレームワークは、せいぜいが「モノづくりをするのであれば、まずはこのような形」とか「ソフトウェアを開発するなら、一般的にはこんなことが必要ですね」というレベルである。本当に役に立つ「知」は、もっと現場に近いところにある。

少なくとも日本人は、フレームワークを考えることが苦手だと思う。むしろ、整理されていない「実践」を多数積み重ね、「底力」を発揮する、というタイプだ。競争優位をいかに維持し高めていくかという戦略を考えたとき、多くの人が「自分の長所、優位な点を伸ばすべき」という。そこで JSQC ソフトウェア部会では、あえてトップダウンの構造を作らず、ボトムアップによる体系化を志向した。

3. 構造・分類と視点

博物学の分類とは異なり、実務における分類には必ず目的が必要である。例えば、乗り物であるか否かという観点であれば、「馬、牛、ロバ」と「ヤギ」に分類できるが、食べ物であるか否かという観点であれば、少なくとも日本では「馬、牛、ヤギ」と「ロバ」に分類できるだろう。このように、構造や分類は、その視点によって複数存在し、どれが正しいかを議論することはあまり意味が無い。むしろ、ある特定の視点から見たときに、最も良い分類は何か、目的に合致した分類は何か、が議論の対象となる。

JSQC ソフトウェア部会では、約 60 の形式知を集め、記述し、それらの体系化をはかろうとした。要求分析から、運用・保守に至る大まかな開発フェーズ、これである程度の分類ができるよね、という

ことは皆合意した。これを「時間軸」と呼んだ。もう一つ、どのような「軸」があるのだろう、と、かなり長い時間かけて議論し、いろいろと試してみた。結局、ひとつの視点に絞り込めなかった。そこで、我々が作成した『形式知集』は『複数』の目次を持っている。目次は、形式知を分類し、並べ替えた項目を示したものであるから、正しく「軸」を意味している。使う状況、読み手のレベルによって異なる目次を参照することになるのだろうか、という想いからである。

4. ソフトウェア部会形式知化プロジェクト生産物の構造

議論の結果『メイン目次』とされたものの表題のみを示す。

1) 物事の真実・原因をつかむ方法を考えたい

- (ア) 現地現物
- (イ) 原因分析
- (ウ) レビュー

2) 対象をモデル化して問題解決の方法を考えたい

- (ア) システム化対象のモデリング
- (イ) 開発プロセスのモデリング
- (ウ) 組織設計のモデリング
- (エ) モデル化と推定

3) 人に動いてもらう方法を考えたい

- (ア) コミュニケーション
- (イ) 育成
- (ウ) 仕組み

現在のソフトウェア開発に置いて重要な「知」は、以下のような三種類に大別できる、というメッセージだと受け取って欲しい。

- (エ) 問題解決（日本的品質管理の考え方・実践）
- (オ) モデリング

(カ) 人

開発対象が硬いか柔らかいかにかかわらず、日本的品質管理における問題解決の考え方や実践は、極めて重要だ。どうもソフトウェアの開発をされている方達は優秀なひとが多いようで、話を聴いて、頭の中で思考実験をして、簡単で有効性がはっきりしたものだけを「やってみたいな」と思う、そういう方が多い気がする。地道にデータを集めるなどというようなことは、どうも苦手ではないだろうか。そのくせ、どこかの雑誌の特集などで「次世代のなんとか」というような特集を見ると、すぐに飛びつく。ちょっと試してみて、直ぐに効果が出ないと、さっさと諦めて次を探す。

もう一度「事実に基づく」という言葉を考えなおしてみたい。これは、「事実」と「感想」が常に分離できる、ということだ。そして、「感想」を裏打ちする「事実」を集めるのではなく、「事実」を説明する「仮説」を懸命に探し出す、という態度だ。一般的に優秀な技術者や管理者は、ある「事実」を見つけると、その「事実」の原因となる問題を、過去の経験などから即座に引っ張り出し、その「原因仮説」を補強する「事実」だけを集めて眺める、という極めて厄介な癖を持っている。これはおそらく、人間そのものが本質的に脳の中に持っている思考パターンだ。それがいつでも自在に使えるからこそ、あまり世の中が変わらないという前提では、問題解決能力が高い優秀な人、と周囲から思われる。このような人は、未知の問題に出会ったときに、いきなり破綻する。しかし、その時には結構偉くなっていることが多いので、その発言には権威があり、組織全体がその破綻にお付き合いすることになってしまう。もう一度「事実に基づく問題解決」を学び直して欲しい。

モデリングに関しては、様々なソフトウェア工学の教科書を、まずは勉強して欲しい。我々が集めた「知」は、その上手な使い方、である。

最後は「人」である。最初の方で述べた「実世界の本質的非論理性」に対処するためには、「人」のコミュニケーションの質向上しか方策がないように思える。コミュニケーションにおける「記号（＝文字や言葉）のやりとり」と「情報・知識のやりとり」を、もう一度考え直し、研究する必要がある。「あれ、よろしく」という一言で、相手が全てを理解し、こちらが期待することをすべてやってくれる、というのはいったい何故なのか。どうすれば、そのような関係になれるのか。逆に、そのような関係にならないことを「お互いが理解する」ためにはどうすればよいか。どのようなコミュニケーションの作法が必要とされるのか。

ある人から尋ねられた。「一緒に飲みに行く」ことが有効だということは皮膚感覚的に分かるし、いろいろな人も言っている。では「一緒に飲みに行かない」ときでも、同じような効果を上げるためにはどうすればよいのか、と。私たちは、二つの方向から問題を解かなければならないのだろう。「仕事の文脈で、どうすれば『仲良く』になれるのか」という方法論、そしてもうひとつは『仲良し』でなくても良好なコミュニケーションを維持する」方法論である。特にオフショア開発の問題などを考える上では、真面目に後者を考える必要がある。

5. 知識伝承の難しさ

近年の開発では、業務が細分化され、それぞれ専門性が高まってきているが、同時に「これは自分の

こととは関係ない」と、他者の経験を「学ぶ」姿勢が希薄になってきている。これは想像力が欠如しただけなのかもしれないが、忙しい現代の開発者に対し、自分にとっての問題であると興味をもたせることが極めて重要である。また、体系化されていない「知」をどのように「教える」のかは極めて難しい問題で、「経験しなければ分からない」というだけでは物事の解決にはならない。

一般的な教育の理論では、新しい事柄を教えるときに、生徒側の情報処理が多ければ多いほど、定着しやすい、と言われている。目で追うよりは、声を出して読む、それよりはノートに書き写す、というように、同時に多くの情報処理が行われる方法のほうが、記憶に定着しやすいということである。最も脳を活発に使うのは「考える」ということだ。何かを教えたいと思ったときに、どうすれば生徒に「考え」させることができるか、が教える側のポイントとなる。

人材育成では、「自分で考えることができる」という目標を立てて欲しい。「考える」ためには「知識」が必要だ。家を建てる時のレンガのようなものだ。しかし、多種多様なレンガを持っていたとしても、家はできない。それらを組合せ、場合によっては加工し、手元にない場合は漆喰を固めて応急的な塊を作ることも必要だ。知識を組合せ、場合によっては仮説を立てることもあるということだ。

それらを組合せ、構造を築かなければならない。その構造がいい加減なら、すぐ崩れてしまう。論理的な思考が極めて重要、ということだ。その前に、そもそもどのような家を立てるのか、事前にイメージできていなければダメだ。言い換えれば、どのような問題を解決しようとしているのか、目的が見えていないと論理を組み立てられない。

最後のメッセージは、結局「考えろ」という言葉になった。普段の暮らしでは、あまり考えなくても済むようになってきている。脳は、そのように、多くのことを習慣化して省エネ、はやりの言葉で言えばエコなのである。しかし、新しい状況や問題に直面したときには、「習慣化された行動」だけでは解決できない。その時には一気にエネルギーを投入して「考える」しかないのだ。その時には大量のエネルギーを消費するかもしれないが、良い答えを得ることができたのなら、その後はまた省エネで進むことができる。

部下に「考える癖」をつけてあげること、これが最も優れた人材育成の方針ではないか、と思う。

6. 宣伝を兼ねて

今回部会でまとめた「形式知集」は、3MB 程度のエクセル形式のファイルに仕上げた。目次からリンクをたどって、具体的な「事例」にたどり着くことができる。それぞれの「事例」には「小説」をつけた。イメージとしては、4コマ漫画の原作のようなもので、読者にとって「自分の身の回りでしばしば起きている出来事」のような印象を与えることを意図している。少し前に書いたように、ソフトウェア産業の人々は、黒船のご威光には簡単にひれ伏すけれど、隣人の経験談はあまり聞いてくれない、という危惧があり、少しでも興味を持っていただくためのきっかけを与えたかったからである。

この「形式知集」は、日本品質管理学会に入会し、その上でソフトウェア部会に参加することで無料で入手できる。ソフトウェア部会では、おおよそ1ヶ月に一回、会合を持ち、議論を進めている。現在のテーマは、いわば「遺言プロジェクト 2.0」とも言えるもので、暗黙知の形式知化を見据えた活動を

始めたばかりである。「学会」と聞くと、敷居が高いというような感覚を抱く方も多いようだが、極めて泥臭い話題で、わいわいやっている。部会メンバーの紹介であれば、一回に限り「お試し参加」ができる、という仕組みも作った。この原稿が公開されるのは8月の末ごろだと思うが、時期的にもちょうど良い。日本品質管理学会の「年度」は、様々な歴史的経緯から、10月～翌年9月である。今、入会申し込みをしていただくと、丸々一年間、活動に参加することができる。

それでも踏ん切りがつかない読者のために、「形式知」の一般公開版を差し上げることにしよう。「形式知」集がどのような感じのものなのかを知るために、3事例だけを掲載した「お試し版」である。掲載事例が3個しかないことを除けば、元の「形式知集」と同じ構造をしている。おおよそどのようなことが書いてあるのか、是非ご自分の目で確かめてみて欲しい。

興味を持ったら、ソフトウェア部会へご参加を。新しい人が一人加わることで、視野が広がり、議論の範囲も広がっていく。お待ちしております。

[ソフトウェア部会 遺言 PJ 形式知集\(一般公開版\)](#)

[エクセルデータダウンロードはこちら \(960KB\)](#)

プロフィール

兼子 毅 (かねこ たけし)

1985年東京大学工学部反応科学科卒業。1991年東京大学大学院工学系研究科博士後期課程修了（工学博士）。東京大学助手を経て、武蔵工業大学（現・東京都市大学）に奉職。ソフトウェアの品質管理、知的労働従事者の動機付け、自然言語からの情報抽出などの研究に従事。（社）日本品質管理学会ソフトウェア部会前部会長。

3. SQuBOK®

■ SQuBOK ガイドの改訂に向けた活動について

(株)NTT データ 技術開発本部
プロジェクトマネジメント・イノベーションセンタ
町田 欣史

はじめに

SQuBOK®ガイドの第1版が2007年11月に出版されてから約3年が経ち、この間にIT業界には様々な変化がありました。身近なところでは、iPhone、Android携帯などのスマートフォンやiPad、Kindleなどの新たなデバイスの登場、Twitter という新たなコミュニケーション手段の流行、クラウドコンピューティングを使った新たなビジネスモデルの確立といったことがありました。このように、ITの技術は日々進歩しています。そして、それらの技術を使った製品やサービスの品質を確保するための技術や考え方も追従して変化していかなければなりません。

SQuBOK®ガイドの目的の一つには、これまで日本で培われてきたソフトウェア品質に関する暗黙知を形式知化するというものがあります。これは普遍的なものであり、技術が進化しても変わることはない品質に対する考えになります。一方で、SQuBOK®ガイドのもう一つの目的として、ソフトウェア品質に関する最新のテーマを整理、体系化することがあります。従って、SQuBOK®ガイドは、新たな技術の登場に伴って変化するソフトウェア品質に関する情報を取り入れて、不定期に更新していく予定になっています。

2008年10月にはSQuBOK®ガイドのアmendメントを [Web で公開](#)しました。ここでは、第1版では対応できなかったパブリックコメントへの対応や、参考文献や論文の最新化を行いました。それからさらに2年が経ち、このたびSQuBOK®策定部会では、SQuBOK®ガイドのメジャーバージョンアップ（第2版）に向けて作業を開始しました。まだ改訂に向けた議論の途中段階ということもあり、確定の情報ではありませんが、検討している改訂内容の一部を紹介したいと思います。

設計領域に関する解説の追加

SQuBOK®ガイドの第1版で扱えなかった領域として、ソフトウェア開発プロセスの上流から中流にあたる、要求分析、設計、実装の部分があります。これらの作業におけるマネジメントや技法については、SQuBOK®ガイド第1版の樹形図には含めていたものの、詳しい解説はペンディングとなっていました。次版では、これらの部分についての解説を含める予定です。これにより、ソフトウェア開発プロセスの全体がカバーされることとなりますので、実装を担当するプログラマーの方、アーキテクチャ設計を行うソフトウェアアーキテクトの方、要求分析を担当する要求アナリストやビジネスアナリストといった、これまでSQuBOKとはあまり縁のなかった方々にも興味をもっていただけることと思います。

ただ、これらの領域はSWEBOK（ソフトウェアエンジニアリング知識体系）でも知識体系として整理されており、内容に重複が出てしまうことを懸念しています。SQuBOK®ガイドは、SWEBOKを否

定や排除するものではなく、その中からソフトウェア品質に関する要素を抽出したものです。従って、SQuBOK®ガイドでは、「品質」という視点で、これらの領域におけるマネジメントや技法について解説していく予定です。

新たなトピックスの追加

SQuBOK®ガイドに記載されている技術や考え方は、業界での認知度もあり、ある程度成熟しているものが基本となっています。ただし、中にはやや新しめのキーワードをあえて載せているものもあります。たとえば「ディペンダビリティ」などはそれにあたるかもしれません。これは、SQuBOK®策定部会として、ソフトウェア品質技術者や開発者の皆さんに知っていただきたいという思いが入っているものです。そういった一部の例外を除けば、目新しい技術などが世の中で一時的に流行しても、まだ技術として確立していないもの、現場に根付いていないものは掲載を見送ることがあります。

たとえばSQuBOK®ガイドの第1版にはトピックスとして「Vモデル」はありますが、「Wモデル」はありません。これは、第1版の執筆当時は「Wモデル」は認知度が徐々に高まりつつあるぐらいの段階であり、まだ開発現場では一般的でないという理由から、時期尚早ということで掲載を見送りました。しかし、この3年の間に「Wモデル」は（適用している、していないは別にして）かなり認知度が高まり、多くの文献にも当たり前のように記述されるようになったと感じています。従って、次版では「Wモデル」を新たなトピックスとして追加することが検討されています。このように、日々新しくなるソフトウェア開発の技術を取り込むことで、SQuBOK®ガイドの鮮度を保っていかようとしています。

規格の最新化

SQuBOK®ガイドでは、付録としてISO、IECやIEEEなどの規格の一覧を掲載しています。それらの規格はトピックスとして解説されていたり、他のトピックスで参照されたりしています。国際標準や業界標準の規格に基づいていることは、SQuBOK®ガイドの記載内容の信頼性を高める一つの要因になっています。しかし、これらの規格は頻繁に改訂されたり、JIS化されたりしています。SQuBOK®ガイドの信頼性を確保するためには、これらの規格は最新のものを参照し、それに基づいた記述に改める必要があります。これも次版での改訂のポイントの一つです。

たとえば、テストドキュメントに関する規格であるIEEE Std 829は、SQuBOK®ガイド第1版では1998年版を参照していましたが、この規格は2008年に改訂されました。そこで、2008年に公開したSQuBOK®ガイドのアmendメントでは、付録の規格一覧を更新し、さらにこの規格を解説、参照しているトピックスの解説文への影響を確認しました（影響はないということで解説文の変更はありませんでした）。今回のSQuBOK®ガイドの改訂にあたっては、改訂のあった規格を確認し、関連する解説文の見直しをする予定です。

終わりに

今回取り上げた以外にも、SQuBOK®ガイド全体を通しての不整合の修正や、第1版で取りこぼしていた情報の追加、読者や有識者からの要望やコメントの反映などを行って、より充実したガイドに改訂

する予定です。

改訂版の出版は来年 2011 年の 1 月頃を予定しております。是非楽しみにしてください。

※iPhone、iPad は Apple Inc.の商標です。その他本文中の製品名は各社及び商標権者の登録商標あるいは商標です。

プロフィール

町田 欣史（まちだ よしのぶ）

(株)NTT データ 技術開発本部 プロジェクトマネジメント・イノベーションセンタ

JSTQB テスト技術者資格認定 技術委員会

社内にてテストプロセス改善の研究、テスト自動化支援ツールの開発・導入支援、ソフトウェア品質向上のための社内サービス運用を行っています。

4. トピックス (1)

■ (1) 『ソフトウェア品質保証部長の会』 活動状況の報告

日科技連 SQiP 事務局

執筆協力:孫福 和彦

(株式会社日立システムアンドサービス)

■活動の背景

近年、ソフトウェアに起因する品質事故が増えており、ソフトウェア品質保証部門の担う役割はどんどん大きくなっています。それに伴い、ソフトウェア品質保証部門の行うべき活動は多岐に亘っていますが、様々な状況のプロジェクトに対し「最適な品質保証活動はどうあるべきか?」、日々悩みを抱えているのが実状です。そこで、2009年11月よりソフトウェア品質部長の会を発足し、ソフトウェア品質保証部門の部門長によるセミナーやグループディスカッションを通じ、品質保証のあり方や事例の作成、開発現場への普及を推進しています。今回はこれまでの活動状況と今後の予定などをご紹介します。

■活動の目的

ソフトウェア品質保証における現場での悩みや意見交換、品質保証の実践的なノウハウなどの議論を通して、品質管理手法や具体的に適用する際の助けとなる事例を蓄積する。

■これまでの活動内容

昨年11月より、ソフトウェア関連の品質保証部長、総勢21名が月に一度集まり、ソフトウェア品質について熱く語り合っています。ひと口に“品質保証”と言っても、その活動内容は各社さまざまであり、一堂に会することで、現場での悩みや意見交換、品質保証の実践的なアプローチなど、会社の壁を越えた活発な意見交換が行われています。

現在は8月末に開催されるSQiPシンポジウムでの発表に向け、みなさん忙しい中、資料作成が大詰めを迎えているところです。

これまでの活動状況は以下の通りです。

- ◆第1回(11/17(木)) キックオフと問題提起(講師:西 康晴氏(電気通信大学))
- ◆第2回(12/17(木)) メトリクスに関する講演とディスカッション(講師:野中 誠氏(東洋大学))
- ◆第3回(2/4(木)) 品質の考え方に関する講演とディスカッション(講師:飯塚 悦功氏(東京大学))
- ◆第4回(3/4(木)) 原因分析に関する講演とディスカッション((講師:金子龍三氏((株)プロセスネットワーク))
- ◆第5回(4/13(火)) 品質保証全体像に関する講演とディスカッション(講師:保田 勝通氏(つくば国

際大学))

◆第6回(6/8(火)) グループディスカッション

◆第7回(7/6(火)) グループディスカッション

■今後の活動予定

活動の成果をまとめ、セミナー等で発表し、順次情報発信していく予定です。

・ソフトウェア品質シンポジウム2010 (SQiP2010) (開催日:2010年8月27日(金))
=>これまでの活動内容をまとめて発表します。

・発表会 (開催日:2010年11月29日(月))
=>シンポジウムの後、内容の補足や次年度への計画など、さらに充実をはかって発表します。次年度の参加者への引き継ぎが中心となりますが、一般参加も募集しますのでよろしくお願ひします。

【発表の概要】

ソフトウェア品質シンポジウム2010 (SQiP2010) での発表概要は以下の通りです。
テーマ別に3つのグループに分け、ディスカッションした内容をまとめて発表します。

■グループ1

テーマ 「SQiP ソフトウェア品質保証実態アンケート2010 (プレ調査版)」

「SQiP ソフトウェア品質保証部長の会」から、国内企業におけるソフトウェア品質保証活動の実態と、その傾向のアンケート調査結果を紹介します。アンケートでは、品質保証組織の体制、役割、活動内容、定量的管理への取組み状況などを調査しました。シンポジウムでは、部長の会参加企業を対象としてプレ調査版としての調査結果を紹介します。皆様の品質保証活動を企画・推進するにあたって役立つ情報となれば幸いです。

■グループ2

テーマ 「各社の品証部長が語るソフト品質保証の取組み」

近年、ソフトウェアの品質の重要性がますます高まっています。品質に関する会社の中核組織である品証部門の長が集まった「ソフトウェア品証部長の会」では、高品質のソフトウェアを出荷／提供するための取組みについて議論しました。本シンポジウムでは品質保証に関する開発工程毎の取組みの傾向や、特徴のある活動について紹介することで参加された皆様の、品質向上の取組みの一助としていただきたいと思います。

■グループ3

テーマ 「品質保証部最前線～オジサンたちも悩んでいる～」

私たちは品質保証部の長である、という一言で括れないほど、部署の規模も権限も品質保証の対象も異なる、様々な立場の部長が集まりました。活動はそれぞれの悩みを共有することからスタートし、昔からあるもの、今だからこそ起きたというものまで、多くの悩みが洗い出されています。今回は尽き

ぬ悩みの中から、「人」、「組織」、「保守開発」という3つのテーマに絞り、現場で起きている問題と、私たちがとっている対策について発表します。

■次年度の会員募集

本活動は、毎年11月を起点に1年サイクルで活動していきます。

次年度の参加募集は10月に行う予定です。

より多くのソフトウェア品質保証部長の方々とネットワークを作り、活発な議論を推進していきたいと思っておりますので、
よろしく願いいたします。

【参加者のメリット】

◇品質保証活動に関する意見交換や情報交換ができます

◇社外のネットワーク作りができます

◇豪華な講師陣による講演・議論ができます

プロフィール

孫福 和彦（まごふく かずひこ）

株式会社 日立システムアンドサービス デジタルメディア2部

SQiP ステアリング委員会委員

ソフトウェアテスト技術者認定(JSTQB)技術委員

ソフトウェア品質のマネジメントや技術(プロセス QA、静的解析、テスト、品質分析・評価など)の研究、および開発現場への展開と定着化を推進

4. トピックス (2)

■ ソフトウェア品質技術者資格認定 中級試験スタート！ ～ソフトウェアの品質技術を高め、継続的・効果的に品質向上を目指す～

財団法人 日本科学技術連盟

SQiP 事務局

1. いよいよスタート 中級試験！

このたび、ソフトウェア品質技術者資格認定試験（資格認定に関しては参考をご覧ください）として、中級資格試験を2010年11月6日にスタートします。中級は、ソフトウェア開発プロジェクトを品質面で成功に導くことができる技術レベルを目標としています。試験対象とする方は、品質保証技術者はもちろん、プロジェクトリーダー、設計技術者、実装技術者、テスト技術者、保守運用技術者、教育訓練職、企画職、営業職、マネージャ層、経営層といったソフトウェア開発に関わるすべての方々です。

2. 中級試験の概要

- (1) 年1回の定期試験
- (2) 試験時間：2時間（休憩なし）
- (3) 試験問題 ※サンプル問題は Web サイトへ掲載予定しています。
 - ① 選択式 25問
 - ② 記述式 3種類の問題形式（下記）よりあわせて15～20問程度
 - ・ 文章中の用語の穴埋め：10問程度
 - ・ 用語についての定義や活用方法の説明（50字程度）：5問程度
 - ・ あるテーマについてその理由や留意点などの考察を記述（200字程度）：2問程度
- (4) 合格ライン：70%程度（選択式、記述式とも）
- (5) シラバス (<http://www.juse.or.jp/software/35/#02>) に準拠して出題

(参考図書)

主参考図書

ソフトウェア品質知識体系ガイド –SQuBOK Guide-

著者 SQuBOK 策定部会 編

出版社名 株式会社オーム社 (ISBN 978-4-274-50162-3) 発行年月 2007年11月

副参考図書

ソフトウェア品質保証入門 –高品質を実現する考え方とマネジメントの要点

著者/訳者名 保田勝通/著 奈良隆正/著

出版社名 日科技連出版社 (ISBN : 978-4-8171-9263-9) 発行年月 2008年04月

さあ、みなさん。中級資格にチャレンジしてみませんか？受験資格は特にありません。
多くの方々の受験をお待ちしています。

[参考]

ソフトウェアの品質向上に関する知識を効率的に身につける手段としての資格認定制度

私たち SQiP は、ソフトウェア品質に関する資格認定制度「ソフトウェア品質技術者資格認定制度」を 2008 年 12 月に創設しました。上級・中級・初級のうち、現在は初級のみ実施されています。

資格認定制度の概要は <http://www.juse.or.jp/software/35/> をご覧いただきたくと思いますが、毎回、定員を大幅に上回る受験申し込みがあり、改めて、近年の品質確保への強い要求を実感しています。

資格認定制度は、ソフトウェアの品質向上に関する知識を効率的に身につける一つの手段として利用いただきたいと考えていますが、以下のような様々な立場の方に有効です。

- 品質向上に関する知識が不足している方 ……

資格認定制度をきっかけとして包括的な知識を体系的に学んでください！

- 既に十分な知識を有している方 ……

自分の知識を体系的に整理して自信を深め、さらなる活用に励んでください！

- 組織を率いる立場の方 ……

組織の持つ知識の底上げによって効果的な品質向上を行えるよう体質改善を始めてください！

この資格認定制度は前述のようにあくまでも手段です。大事なことは、ソフトウェアの品質向上に関する包括的かつ体系的で本質的な知識を身につけ、効果的な策を継続的に実施することです。多くの組織や多くの技術者・管理者・経営者が、この資格認定制度をきっかけとして品質向上が実現できれば、これほど嬉しいことはありません。

資格試験の詳細はこちらから

(9月中旬に試験のご案内の更新をいたします) ⇒ <http://www.juse.or.jp/software/35/>

(1)各試験の特徴

ソフトウェア品質技術者資格制度は、初級、中級、上級の3段階の資格により構成されます。上級の試験については今後新設して行く予定です。

- ①初級 ソフトウェア品質知識を知っており、知識を使えるレベル（開発プロジェクト内のレビューやテストといったタスクを確実に遂行できるレベル）。
〔ITSSレベル1~3相当 選択問題 40問〕
- ②中級 <2010年11月実施>ソフトウェア品質知識を詳しく理解し、現実フィールドで応用できる（プロジェクト全体のソフトウェア品質確保をリードできるレベル）。〔ITSSレベル3~4相当 選択問題 + 小論文〕
- ③上級 <今後検討予定>ソフトウェア品質知識を熟達しており、実務に応用できる（ソフトウェア

開発組織のソフトウェア品質向上をリードできるレベル)。

[上級 ⇒ ITSSレベル5~相当 論文 + 業務経歴書 + 面接]

5. 健康

■ ポジティブアプローチ ポジティブ心理学をベースにしたイキイキした職場作りと生産性の向上の手法

一般社団法人ポジティブイノベーションセンター
代表理事 渡辺 誠

1. はじめに 今のソフトウェア開発の課題

最近、ソフトウェア開発部門の経営者や現場のマネジメントの方々からこんな事をよく聞くようになりました。

「よりよい品質の商品を低コストでしかも短期間で作る高い生産性が求められているのです。そのためには、SQC やソフトウェア開発のさまざまなツール・手法などを勉強し、専門的な技術アプローチもいろいろ取り入れてきたのですが、いま、それでは、間にあわなくなっているのです。専門技術を高度化することはこれからもやっていくつもりですが、社員一人ひとりのモチベーションを高くして、生産性を上げることはできませんでしょうか？ソフトウェアはやる気になって作ったものと、作業として漫然と作ったものでは品質が格段に違うのは、みんな分かっていると思うのです。人の心に働きかけることにより、より高いパフォーマンスを達することはできるのでしょうか？」

皆さん、とても真剣です。

こんなことができるのでしょうか？あまりにも理想かもしれません。しかし、これはできるのです。「ある役員の方は、これができたら、最後の砦かもしれないね。」と言っていました。

それを実現するにはどのようにすればよいのでしょうか？

ポジティブ心理学は「人の力を最大限に引き出すこと」を研究する学問です。1998年にアメリカから始まり、ヨーロッパに広がりました。いまは、シンガポールや中国でも盛んに研究され、応用されるようになりました。その研究成果を使った方法がいくつかあります。その中から今回はポジティブアプローチを紹介しましょう。主に、マネージャーが部下の力を最大限に引き出すという視点からご紹介したいと思います。

2. 今までの考え方とポジティブアプローチ

我々が進めてきた、科学的管理法は素晴らしいものがあります。これが日本の高品質を作り出し、支えてきました。日本の誇るべき考え方であり、今産業界の中心的考え方です。

我々が作っているシステムや商品に欠陥があってはなりません。製品は、完璧に作動することで信頼性が得られます。したがって、欠陥があれば、真の原因を見つけ出し、根本的な解決策を見つける必要があります。そこで、科学的管理法に基づいた品質管理が大切になります。データを基に、なぜなぜ問答を5回繰り返して本質的な問題を見つけ、解決するのです。ソフトの現場にいる方には言うまでもありませんね。我々がとても慣れている考え方です。そして、高品質の商品を作り上げる根本的な考え方です。この考えは、製品のように完璧を求められるものに利用されると効果的です。

しかし、この手法が人のマネジメントに利用されるとどうなるでしょう。

「なぜ、できないのだ。」「本当の原因は何なのだ。」「これが欠けている、これができていない、原因は・・・」と詰めてもいい答えは出てきません。言われた人はどのような気持ちになるのでしょうか？その人の気持ちは落ち込み、自信をなくして、エネルギーが落ちます。自信が無い状態やエネルギーが落ちた状態を繰り返しつづけると、鬱などの病気になるかもしれません。

人は完璧ではありません。完璧でないものに対して、完璧を求める問題解決手法を適用しているのです。弱みを矯正するより、強みを使ったほうが成果を挙げるのが人間です。

そして、その強みやうまくいく仕事の仕方は人それぞれ違います。金太郎飴のように全く同じではないのです。

あまり、ここでは、テーマが異なるので詳しくは述べませんが、ポジティブ心理学で証明された事を実践している活力のある職場では、鬱などの燃え尽き症候群が極めて少ないことが研究の成果から明確になっています。

3. ポジティブアプローチとは

今までの、「何ができていないのだ。」「なぜ、できないんだ」という問題志向の考え方を変えたらどうなるでしょう？

「ゴールはどのようにしたら達成できるのだろうか？」という発想に変えて見てください。当たりまえのことを言っているようですが、ありたい姿を明確に描き、それを達成することを目標におき、マネジメントをすると、その結果には大きな違いが現れます。

それを実現するのが、ポジティブアプローチです。図1をご覧ください。これがポジティブアプローチの概要です。



図-1 ポジティブアプローチ

ポジティブアプローチでは以下の事を考えます。

Step0 強み

まずは、仕事をチームで実施する際に「強み」をベースに役割を検討します。「強み」を活用できる役割をアサインすることにより、メンバーの継続的なやる気を高め、高い生産性を確保します。これは、仕事が始まるステップに入る前の大切なマネージャーの実施項目です。

その後、目標を達成するために、以下のプロセスを適用していきます。

Step1 ありたい姿(ゴール) :

ありたい姿を明確に描き、それを実行する意味をメンバー間で共有します。この時点で、仕事をしている意味にメンバーの共感を得られれば、メンバーのベクトルが合います。この目的に集約されます。

Step 2 よいことさがし

「うまくいっていることはなにか?」「なぜうまくいっているのか?」を探求します。うまくいっていることは継続します。そして、このうまくいった事をつくりあげたのには、どんな努力や工夫があるかを探しだし、メンバーの理解と動機付けをします。メンバーがすごい事をしたと思ったら素直に気持ちを伝えてあげるのが、さらに動機付けの効果をもたらします。

Step 3 うまくいかないこと

「うまくいかないことはなにか?」を、明確にします。同じ事を繰り返しても違う結果は得られません。「うまくいかないこと」を明確にして、違うことを実施するためのヒントを探します。

Step 4 次の手さがし（選択肢）

「うまくいったことを継続するのはどれか?」「うまくいかないことをやめて、何をすればありがたい姿につながられるのか?」を検討します。できれば他のメンバーにも参加してもらい選択肢を検討するとチームがまとまります。

Step 5 次の一歩

「具体的なアクションは何をするのか?」を明確にします。次回、(来週か 2 週間後)に報告してもらうことを約束します。勇気付けて現場に戻ってもらいます。

4. ポジティブアプローチの効果

ポジティブアプローチは図-2のような考え方をベースにしています。ポジティブな考え方が根底にあることがお分かりいただけますでしょうか？この考えをベースに会話を進めると大きな違いを作れます。

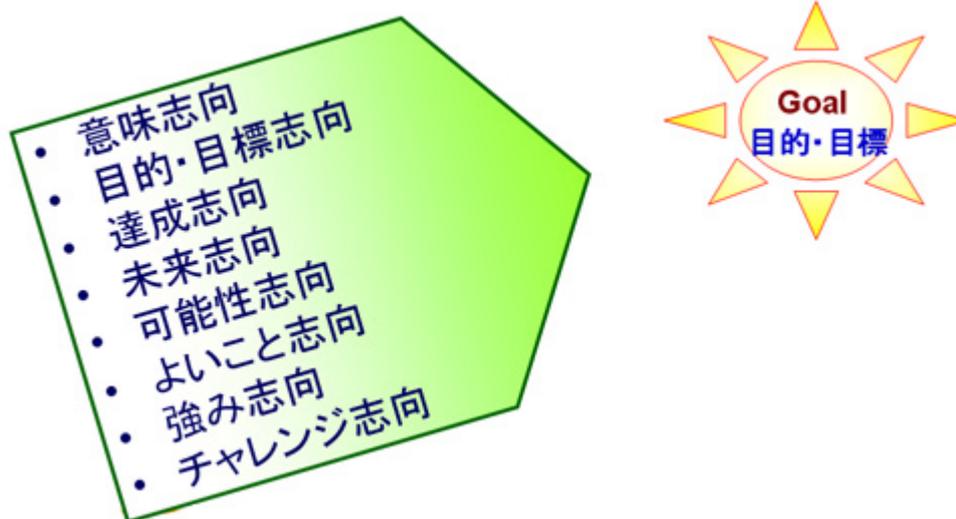


図-2 ポジティブアプローチにおける基本的な考え方

このポジティブアプローチは人に自信をあたえます。人が継続的に積極的に仕事に関わるための秘策がここにあります。各ステップでどのようなことが起こりうるのか、一つひとつ見て行きましょう。

Step0 強みを活用する効果

強みとは ①、うまくできること、②、エネルギーのでること、③、よく使うことです。ただ、うまくできるだけでは強みとはいえません。その人が時間を忘れるほど夢中になれる。それを使って仕事をしているときにワクワクする。そんなときに人は充実します。その充実感をつくってくれるのが、その人の強みです。

この強みを使って仕事をしていると、短い時間で高い成果をあげ、成功確率が高くなるのがポジティブ心理学の研究成果で証明されています。強みをたくさん使うことができれば、その人にとってその組織は、やりがいのある意味のある組織になるのです。

マネージャーは人の強みをよく見ていく必要がある時代になってきました。人の強みは一人ひとり違います、その「違いがあって当たり前」という発想を、頭だけでなく、自分が本当の心の深いところで理解できたときに、人を活かしかれるチームマネジメントができます。

Step1 ありたい姿の効果

メンバーのベクトルを合わせるための一番大事な作業です。マネージャーの思いとメンバーの思いが違

うことはどこの現場でも良くおきています。案外、プロジェクトをすすめていて、目的が合っていない仕事をしている人が多いものです。意味を説明し、仕事を達成した後で得られる効果をみんなが納得すると、ベクトルが合い、チーム力を最大限に発揮することができます。メンバー全員が絵を書いたようにありたい姿を明確にイメージできるようになることがポイントです。

Step 2 よいことさがしの効果

うまくいっていることを大事します。うまくいっていることは繰り返して行い、さらに熟達させていきます。うまくいったことについてなぜうまくいったかの原因を聞いていくと、メンバーの強みや大切にしている考え方が分かってきます。

できれば、マネージャーがメンバーの方に「うまくいったことは何ですか?」「あなたが何をしたらうまくいったのですか?」という質問をしてくれるとメンバーのやる気があがります。そして、強みや自分の大切にしている価値観に気付くことは、これから先の継続的なやる気の維持に繋がります。

また、メンバーにとってはうまくいったことを明確にしていくプロセスで、自分が工夫し、努力してきた結果を理解してもらうことができます。人にとって、自分の努力や工夫を理解してもらうのはとても大切なことです。自分を理解してもらい、認めてもらうことによって自信を得ることができます。

Step 3 うまくいかないことへの対処

うまくいかないことについてもあまり失望はさせません。失敗したそこから何を学ぶかという姿勢が大事にします。そして、失敗から学びを得たら、ありたい姿の未来に向かって、どうすればよいかと考え前にすすめていくのです。この姿勢も、職場にポジティブな風土をつくります。

この考えを進めるには、前提として、成功の裏には失敗があることを認めなければなりません。失敗は悪というような、減点主義の組織風土ではこの考えを浸透することができません。そんな組織には、どんな成功者でも、失敗の経験の上に、今の成功があることを意識付ける必要があります。

Step 4 次の手さがし（選択肢）の効果

問題を解決するには、目標やありたい姿に向かって、いかに多くの選択肢を出せるかが課題です。ここでは、どのような方法があるのかを、たくさん出します。できるだけチームで考えましょう。ブレインストーミングなどの手法を使うことが多くあります。

創造性については、以下のことがポジティブ心理学では研究成果として発表されています。「ポジティブ感情が豊かな日にはより創造力が発揮される。」ポジティブ感情とは、充実・感謝・親和などがありますが、自分に対する肯定感と言っても良いかもしれません。自己肯定感をもてない人は、アイデアが出にくだけでなく、出したアイデアも自信が無いので、採用されることが少なくなります。そして、なおさらアイデアが出にくくなります。アイデアを多くだせる雰囲気をつくるためにも、今までのポジティブアプローチで説明してきた、「強みを活かす」「よいことさがし」「うまくいかないことへの対処」などが大切になるのです。

Step 5 次ぎの一步の効果

次の一步ではすぐできることや、すぐに手をつけられることから始めます。すぐできることですから成功の確率が高くなります。したがって、成功して自信をつけることができますし、この成功の積み上げが自己肯定感を強くさせます。マネージャーや周りの人からの承認や賞賛も得やすいし、本人もプロジェクトが進むので楽しくイキイキと仕事を進める事ができます。

勿論、難しいことにチャレンジもします。その際は、なるべくプロセスを細分化して小さな目標を達成していくようにすると進歩が分かりやすくなります。

5. ポジティブアプローチを利用した人の声

ある IT 企業でポジティブアプローチ研修やポジティブを定着させるための仕組みづくりを実施しました。その企業では教育参加したマネージャーに対して 2 ヶ月間の活動を実施してもらい、部下が評価する仕組みも取り入れました。どのような変化があるかを事前・事後で測定したのです。その結果、チームの絆の強化・ゴール・目的意識の強化、チームの活性化の強化、メンバーの動機付けの強化、について統計上からも明らかな違いが出ました。今回の 2 ヶ月間の測定では、生産性とコミュニケーションは、多少よくなりましたが統計上明らかな効果とは言えませんでした。

その企業のあるマネージャーは「うまくいったことは?」「それがうまくいったのには君のどんな強みが活かした?」とメンバーに問いかけて、部下の理解と強みの発見を同時にしていました。そのマネージャーは「これは魔法の言葉ですね」といっていました。自由な明るい雰囲気ของทีมをつくり、メンバーのやる気を維持するための魔法の言葉だそうです。ポジティブに人に焦点を当てたときに、大きな違いを作り出すことができたとのことでした。

また、他のマネージャーは、「メンバーがいままで比べ、小さな事を相談してくれたり、問題を早めに教えてくれたりするようになりました。」と伝えてくれました。「これにより、プロジェクトを実施する際に、早い段階で手を打てることになり、大きな問題を事前に防ぐことができている」と喜んで教えてくれました。

6. ポジティブアプローチの実現

ポジティブアプローチの考え方は簡単ですが、組織に定着させるには、仕掛けが必要です。まず、自分やメンバーの強みを知るにも仕掛けが必要です。また、ポジティブアプローチを組織に定着させるためにも仕掛けが必要です。

また、問題志向との共存や使い分けも大きな課題です。組織全体が、問題志向もポジティブアプローチも両方使えるようになる必要があります。問題志向は「もの」や「システム」に利用し、ポジティブアプローチは「人」に利用する事をお勧めしています。「人」が「システム」を作っているときは、問題志向とポジティブアプローチの両方を利用すると良いでしょう。

この情報が、皆様の職場がイキイキとして、高いパフォーマンスを誇れるようになることにお役にたてたらうれしいです。さらに、詳しい情報が必要な場合は以下へご連絡ください。

info@positiveinnovation.org

プロフィール

渡辺 誠 (わたなべまこと) (Max 渡辺)

一般社団法人ポジティブイノベーションセンター代表理事

サクセスポイント(株) 代表取締役

富士ゼロックス(株)・富士ゼロックス総合教育研究所にて営業教育・人事教育を企画・開発・実施。人の成長に貢献する事を夢にサクセスポイント(株)と一般社団法人ポジティブイノベーションセンターを設立。ポジティブ心理学を応用し、働く人々がイキイキと最大限の機能を発揮できるようになり、組織に高いパフォーマンスをもたらす事を使命に活動している。国内外のポジティブ心理学者のネットワークも広く、国際的なネットワークの中で新しい手法の研究や開発に取り組んでいる。

一般社団法人ポジティブイノベーションセンターの詳細は以下を参照

<http://positiveinnovation.org/>