Quality One

クオリティ　ワン

Vol.14　2011年5月号

Software Quality Profession

財団法人 日本科学技術連盟

1. 品質

■ プロダクトライン開発における品質の向上効果

セイコーエプソン株式会社
島 敏博

前回の記事では、株式会社エクスモーションの山内 和幸さんによって、ソフトウェア・プロダクトライン(以下、プロダクトラインとします)の基本となる考え方について説明していただきました。プロダクトライン開発ではまず、変化にどこまで対応するかスコーピングを行い、次に可変性分析を行って共通部と可変部を分けた設計に変え、それらから導出して製品を作り出します。

今回は、プロダクトライン開発をおこなうことによって、または、いまの派生機種開発にプロダクトライン的な見方を加えることによって、品質面でどのような改善ができるのか考えてみます。

1．管理するソースが減ることによって品質が向上する

プロダクトライン開発とは、ひとことでいうと、共通部と可変部をわけて開発することにほかなりません。しかも世代ごとに、可変部の内部で共通部を見つけて分離し、共通部分を増やしてゆくことが重要です。

もちろん、プロダクトライン開発に変えてゆく間も、既存製品群のリリースは続けなければなりません。既存製品群をリリースしながら、段階的にプロダクトライン開発に移行してゆくイメージは図１のようになります。

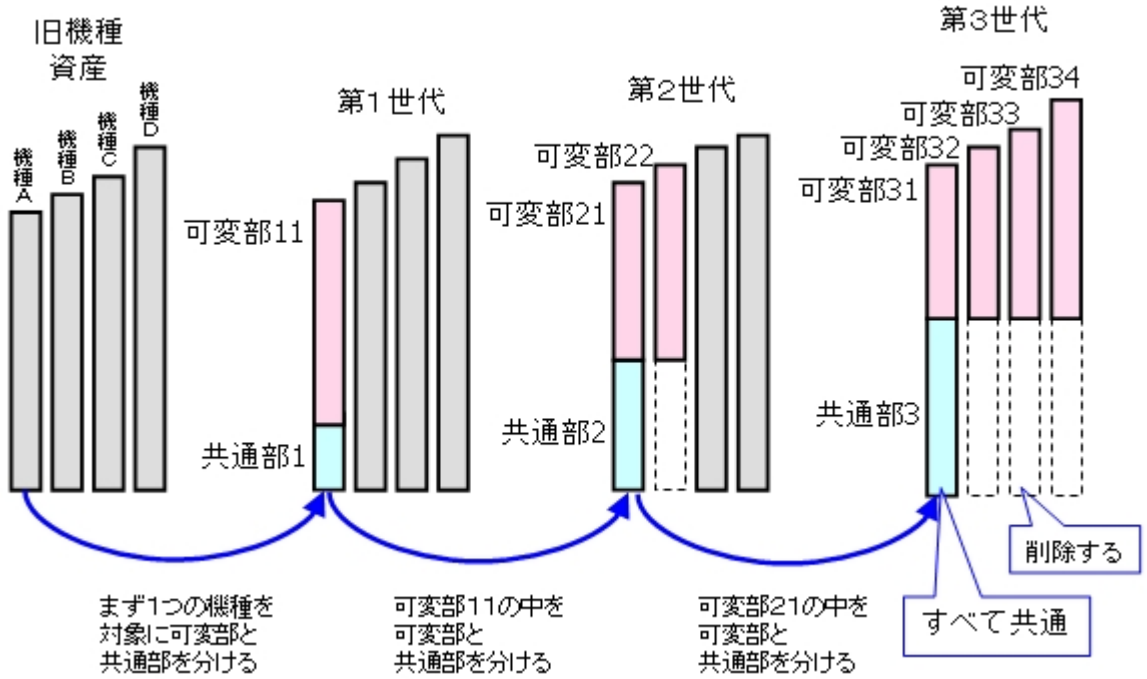

図 1：世代ごとに可変部の内部で共通部を見つけて分離する

図 1 では現在のソースコードを灰色、共通部を水色、可変部をピンク色で表現しています。開発を継続している製品が 4 機種あったとすると、一度にプロダクトライン化することは難しく、まず１つの機種を選択して、それを共通部と可変部に分けます。それを第１世代とします。

第２世代では、共通部を増やすとともに、同じ共通部を使って２機種を開発します。２機種の共通部は同じソースコードなので、ここからソースコード削減効果が生まれます。

さらに第３世代では、共通部をさらに増やすとともに、その共通部を使って４機種すべてを開発します。４機種の共通部は同じソースコードなので、さらにソースコード削減効果が生まれます。

このように、第１世代で、共通部と可変部を分けている段階では、まだソースコード削減効果が出ません。効果が出るのは同じ共通部から複数の機種を作り出せるようになった第２世代以降です。

製品群を生み出すのに必要なソースコードの量が少なくなれば、それだけで品質向上がねらえます。

２．依存性を局所化して品質を向上させる

プロダクトライン開発では可変性分析をおこない、何かに依存している部分を局所化して設計を改善していきます。たとえば製品ごとに異なる OS を使った、複数の製品からなる製品群を作っているようなところでは、OS に依存している部分を局所化する設計に改めます。図１に OS に依存している部分を局所化したクラス構造を示します。

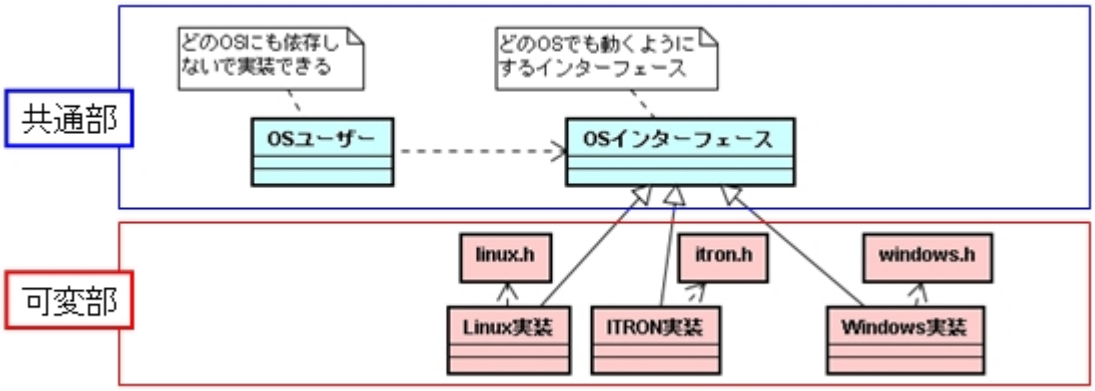

図 2：OS 依存を局所化する

図２では、OS ユーザークラスは、OS インターフェースクラスを使っています。OS インターフェースクラスを実装しているのが、Linux 実装クラス、ITRON 実装クラス、Windows 実装クラスです。

Linux 実装クラスだけが linux.h をインクルードしていて、ITRON 実装クラスだけが itron.h をインクルードしていることに注目してください。このようにすることで、OS インターフェースクラスを使った OS ユーザークラスはどの OS にも依存しないで実装できます。

OS 間の違いによる動作の違いを局所化することで、それ以外への影響を少なくできます。これによって OS 非依存部分の品質を高めることができます。

OS ユーザークラスと OS インターフェースクラスの部分が全ての製品での共通部となります。

このように何かに依存している部分を派生した実装クラスに局所化し、その抽象インターフェースを使ってユーザープログラムを実装することで、ユーザークラスの品質を高く維持することができます。

３．繰り返し検証による品質の向上

プロダクトライン開発では品質が向上するチャンスが２回あります。（図３参照）

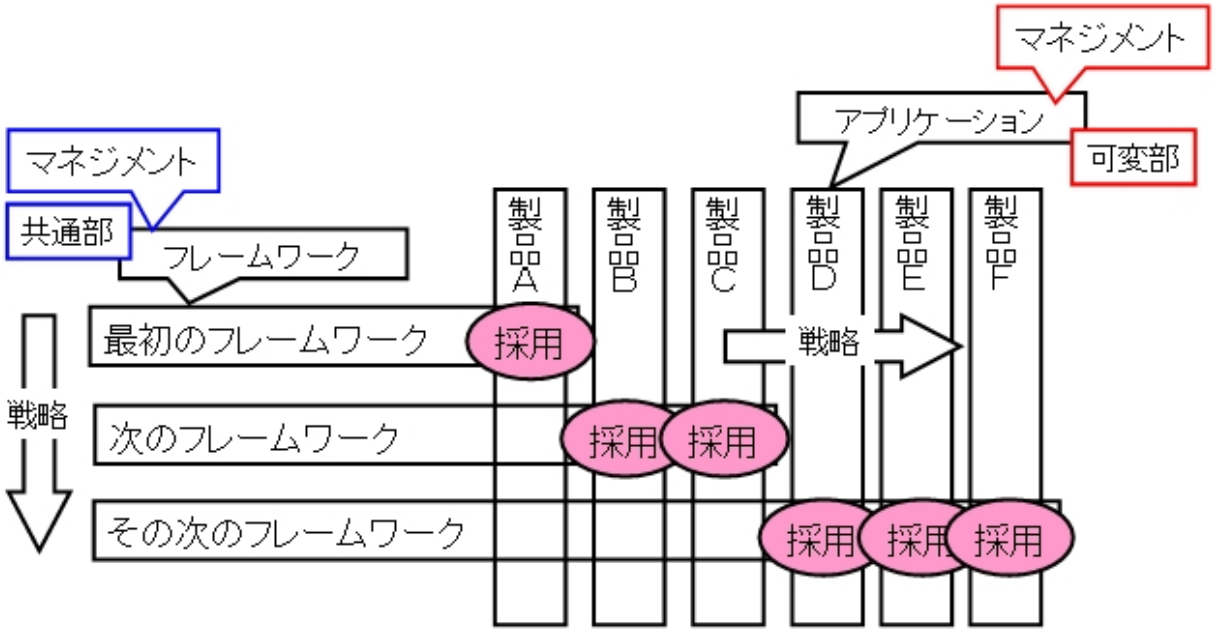

図 3：繰り返し検証による品質の向上

１つめは、フレームワークを作ってリリースするときです。フレームワークチームはリリースするときに仮仕様のアプリケーションと組み合わせてきちんとフレームワークが動作するかを設計者がテストします。

２つめは、フレームワークとアプリケーションを組み合わせて製品開発をするときです。アプリケーションチームはフレームワークを使って製品全体の動きを設計者がテストします。

不具合も局所化される可能性があります。共通部分が他の機種でも使われていて他の機種では不具合が発生していないならば、共通部分よりもその機種用の可変部に不具合がある可能性が高いと言えるでしょう。

コード行数が数百万行におよび巨大なプロジェクトの場合、プロダクトライン化を一度に進めることはできません。そのドメインにとって資産となる一番大切な部分から、順番に共通部と可変部に分ける取り組みを始め、世代ごとに少しずつその範囲を広げてゆきます。

取り組んでいるパッケージの範囲から、品質が向上していく様子をメトリックスに取り、改善を見える化をしてゆきます。第１世代では不具合が局所化されるだけかもしれませんが、共通部が複数の機種に使われる第２世代から繰り返し検証による品質の向上が見込めます。

４．ひとりの担当者が多くの機種を担当することで、ソースの再利用性を高め、品質を高める

機種ごとに担当者が違っているものを、できるだけ同じ担当者で出せるように変えていくのが、プロダクトライン開発です。（図４参照）

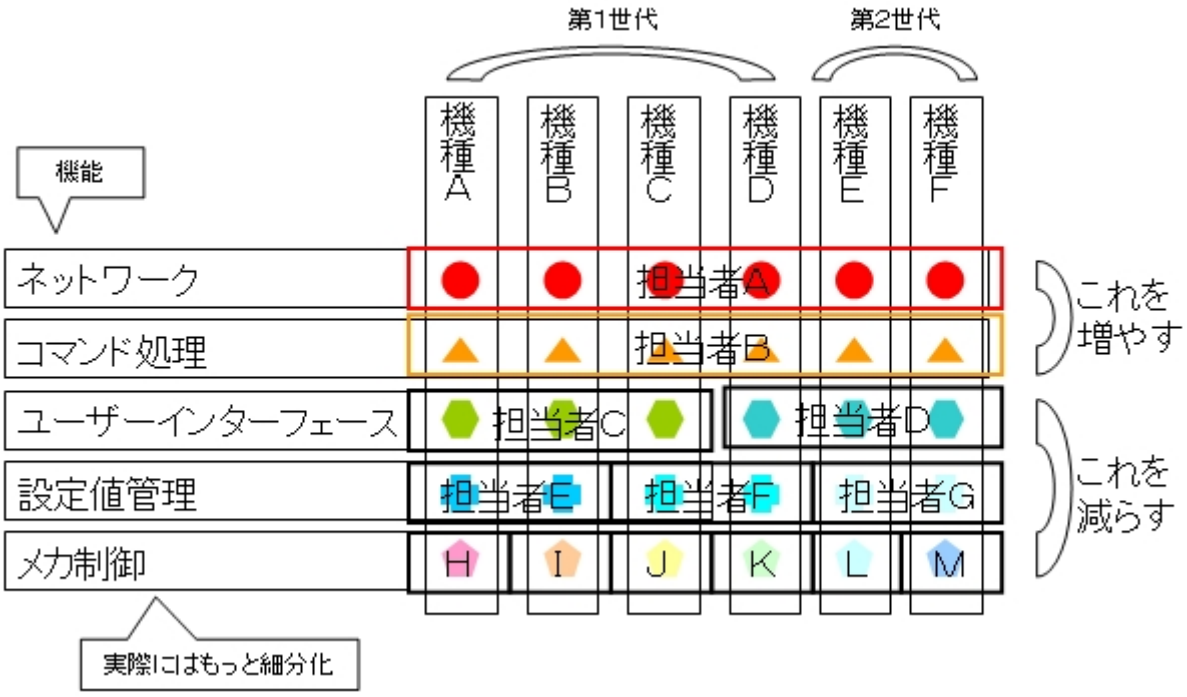

図 4：各機種と各機能が、どの担当者とどのソースからできているのか

機種と機能の関係を図で表したものです。第１世代の機種が４機種あって、第２世代の機種が２機種あります。それぞれにいろいろな機能がはいっています。

プロダクトラインがうまくいっている状態とは、図４の上部のように、どの機種も同じ担当者が作っている状態です。このようになっている部分を世代を重ねるごとに増やしていくのがよい戦略です。

図４では、担当者で分けてみましたが、実際は同じ担当者でも別のソースコードを書いているかもしれないので、正確にはソースコードで分けて考える必要があります。まずこのようなマトリックスを作って、現状の機種と機能がどの担当者のどのソースから作られているのかをまず把握します。

ひとりの担当者が多くの機種を受けもつことで、ソースコードの再利用を意識した作りに変えることができ、コードクローンを少なくし、その機能用のソースコードの品質を向上させる効果があります。

５．機能を担当させる体制づくりにより品質を向上させる

共通部と可変部に分けて開発するのがプロダクトライン開発ですが、これは必ずしも組織を２つに分けることではありません。組織作りにはいろいろな方法があると思いますが、図５のように担当者を決めて機能を担当させる方法がお薦めです。

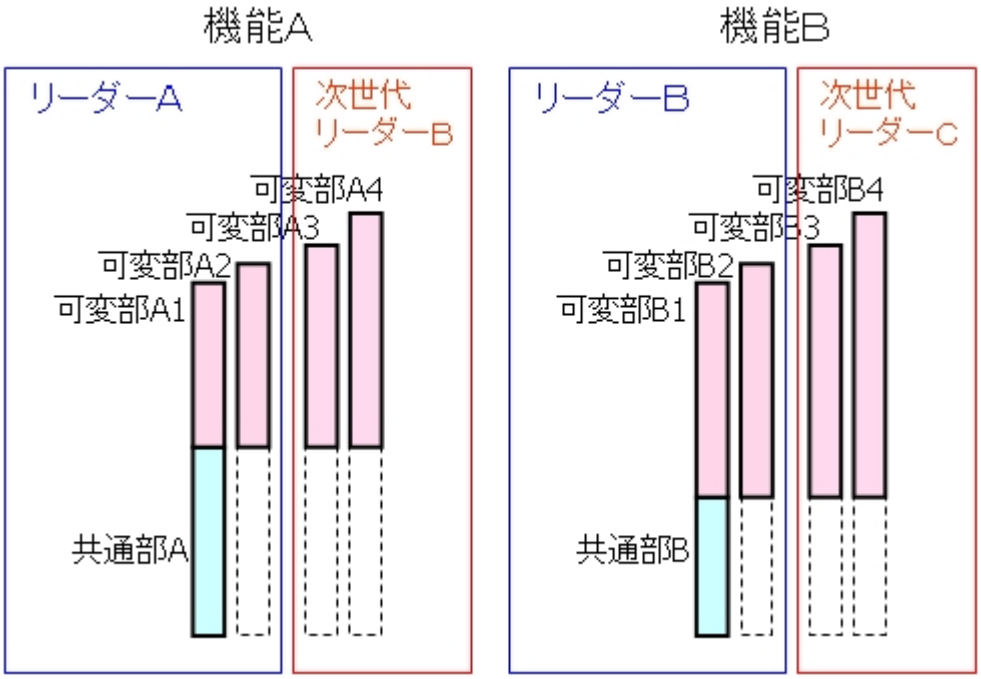

図 5：機能を担当する体制づくり

図５は機能ごとに担当者を割り振る方法です。ある特定の機能Ａに注目すると、その機能の開発に責任をもっているリーダーＡは、その機能の共通部と可変部の 両方を担当しています。第２世代では、自分で作った共通部を使って、２つの機種の可変部を開発し、それぞれ組み合わせて製品をリリースします。

リーダーは共通部を作ると共に可変部を作り、さらに同じ共通部を使って複数の派生機種を開発することをミッションとします。自分で実装しやすくすると同時 に他人にも使ってもらいやすいインターフェースを作るわけです。このようにリーダーは作る側と使う側の両方を担当するということとなります。

さらに派生機種を増やしていく段階で、次世代リーダーＢさんを育成します。そのときはまず可変部を作ってもらいながらプロダクトラインの基本を伝授します。

マスターした次世代リーダーＢは、やがて機能Ｂのリーダーｂとなります。

６．テストもプロダクトライン開発することにより、製品もテストも一括して品質向上させる

モデル駆動開発を活用することで、プロダクトライン開発を加速することができます。製品をモデル駆動開発する場合、テストもモデル駆動開発します。設計したものは開発者が自分でテストします。図６をご覧ください。

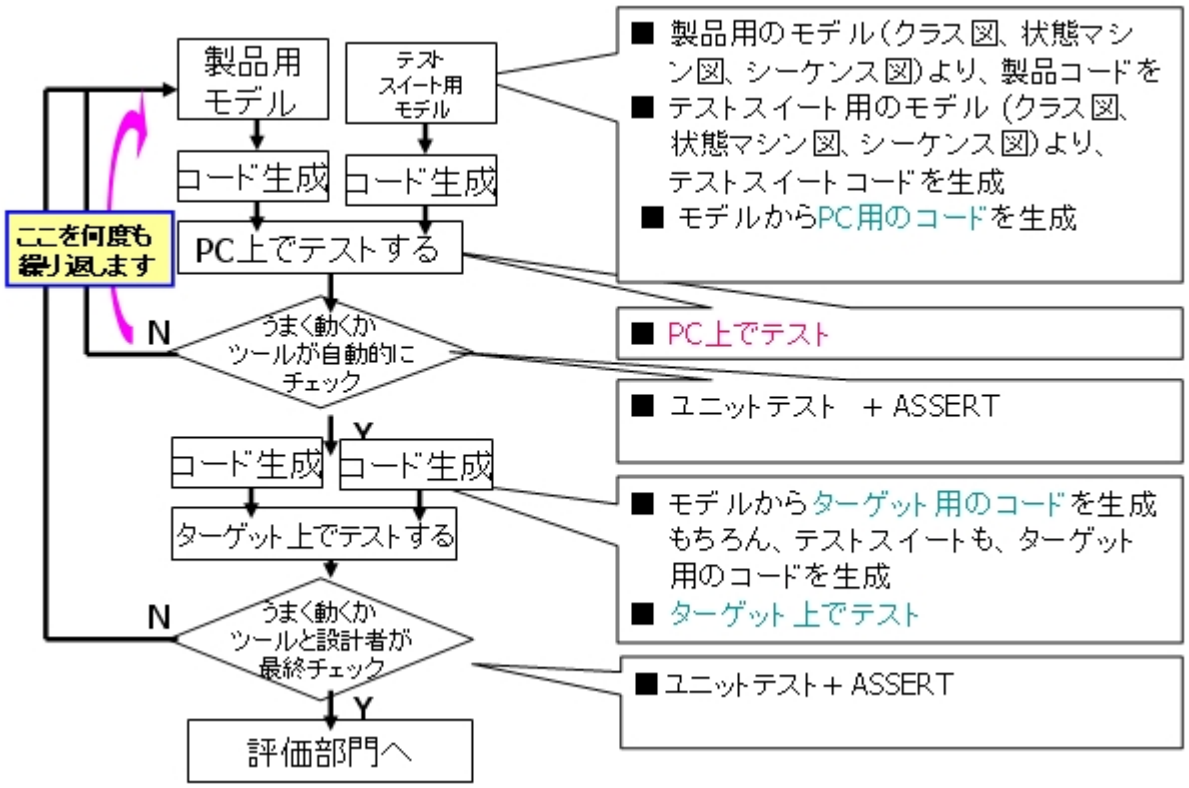

図 6：製品とテストの両方をモデル駆動開発する

設計者は、製品用のモデル図面と、テストスイート用のモデル図面を同時に設計します。クラス図上に、製品用のクラスと、テスト用のクラスを描きます。状態マシン図なら、製品用の状態マシンを表した図のほかに、その状態マシンをテストするためのテストクラス側の状態マシンの図も描きます。

それらの図からそれぞれコード生成させ、PC 上でテストします。

うまく動くかどうか確認しうまく動いたら、今度は両方のモデルから、それぞれ実機用のソースコードを生成させ、ターゲット上でテストをします。

製品コードをリファクタリングした後も、それにあわせてテスト用コードもリファクタリングしてテストが通る状態を維持します。これにより品質が落ちていくのを防ぎます。テスト周期、リリース周期は１日以内とし、常に２つを同時にリリースします。

このとき（テストコード行数／ソースコード行数）のメトリックスをモジュールごとに計測して見える化することをお薦めします。

さらに、いったん作り上げたプロダクトラインプラットフォームが崩れないように監視していくプロ

セスも取り入れます。依存性構造マトリックスを作成し、認められない循環参照、逆向き参照、飛び越し参照などを追加してプラットフォームが崩れ出さないかも監視します。自動化し深夜プロセスに組み込みます。

　このように、プロダクトライン開発では、製品群開発を効率的に進められるように、世代を重ねてよい設計に変え、よいプロセスに変えつづけていくことが大切です。それにより、これまで述べたようにソフトウェア品質を高めることができます。

プロフィール
島 敏博　（しま　としひろ）
セイコーエプソン株式会社
機器ソフトウェア統括センター 機器ソフトウェア企画設計部　研究副主幹
1982 年 信州精器（現セイコーエプソン）入社。以来ドットインパクトプリンタ、熱転写プリンタ、インクジェットプリンタ、レーザープリンタに組み込まれるソフト ウェアの開発設計実装を担当。現在はプロダクトラインを支えるフレームワークのアーキテクチャ設計を担当。モデル駆動開発ツールを使用して大規模組込み開 発を加速中。全社に対してモデリング教育、モデルレビュー、オブジェクト指向の普及、事業ドメインを越えたアーキテクチャ展開を行う。SESSAME 会 員。SESSAME セミナー「モデル駆動開発を組み合わせたソフトウェア・プロダクトライン開発入門セミナー」の講師を担当。

2. 人材育成

■ 勉強会のつくりかた

株式会社 ACCESS フロントエンド事業部
主任 松木 晋祐

このたび「人材育成」というテーマで寄稿の機会を頂きました。私も前回の誉田さんの仰るとおり「人材」は「自らに拠って育つ」ものと考えています。その具体的かつ王道たる手段として「勉強会」があります。

「勉強会」という言葉にみなさんはどんなイメージを持たれるでしょうか？

社内で催される公式/非公式の有志勉強会、社外の技術コミュニティで企画される、そのコミュニティの趣旨に沿ったカジュアルな勉強会、教育事業者が企画 /開発する技術セミナー、赤坂の料亭で催されるという代議士先生らによる政策研究会、などなど。みなさんの背景や所属されている業務ドメインによってその イメージは異なると思いますが、全てに共通するプレーヤーは「主催者」と「講師」と「参加者」です。今回はこのうち「主催者」と「講師」としての立場か ら、実際に開催した「勉強会」を通して学んだ "技術以外の部分" をお伝え出来ればと思います。

「勉強会」を主催する人は何を動機にしているのでしょう。「参加者」として何度も足を運んだ身からしても、日程調整、会場確保、当日の運営、質の担保、場 合によっては懇親会の段取り、などなど大変な労苦が想像されます。そんな催しを実行に移す「主催者」の動機として挙げられるのはおそらく、下記のようなも のではないでしょうか。

(1)ある技術に心の底から入れ込んでいて、ひとりでも多くの方に知ってほしい
(2)偉大な先駆者や身近な先輩に触発されて、伝える側に回ってみようと思った
(3)ひとりで勉強するのは続かないので、みんなを巻き込んで進めようと考えた
(4)自分がわかる技術を教えて、みんなに喜んでもらえるのが純粋に嬉しいから

私もつい先日「上層テスト技術者に贈る Android 開発入門講座」という勉強会を企画し、同時に講師を務めさせていただきました。動機としては (3) に近かったと思います。(参考 http://atnd.org/events/13289)

Android は 2011 年春現在、もっとも勢いのあるモバイルプラットフォームで、国内の主要キャリアから搭載機種が続々と発売されています。しばらくこの勢いは止まりそうになく、普及に従って、エンタープライズのフィールドにおいてもプラットフォームに Android が採用される事例が増えています。そんな中、業務で必要になりそうという状況もあったのですが、そもそも小さいコンピュータが好き、何かその上で動くモノ を作るのが好き、という趣味が半分。勉強会の主題にもあるとおり、上層テスト技術者にも現場で妥当なテスト設計を考える上で、いまどきのモバイルプラット フォームのソフトウェアがどのように動作しているのか、アプリケーション開発者に近いレベルで理解しておいて欲しいという、ここ数年の個人的な思いが半分 で、まずもって自分が学ぶ必要に迫られ本企画の開催へ思い至りました。

【講師として】

なぜ「勉強会」を開催するのが「学ぶ」事に繋がるのか？疑問に思われる方も多いと思います。

ふつう、学んだ後に催すものではないのか。

実は「勉強会」でいちばん「勉強」になるのは講師です。まず、多くの人の前で自分の考えや、その場にいる全員の貴重な時間を使って達成したいことを明確に しなければなりません。次に、講義の途中でなぜこれを教えているのか、どうしてこういう教え方になるのか、を論理的に説明できなければならず、最後に、受 講者の素朴な、思いがけない質問に回答できるだけの一段上位の理解と考察を、前もって準備しておく必要があります。これだけの要件を、明確に示された日時 までに満たすための「学び」は特に効率面において他のどんな方法にも勝るのではないかと私は考えます。

一点、この「学び」に失敗すると参加してくれた方に多大な迷惑を掛けてしまう点において非常にリスキーな方法ではあります。しかし、成功すれば自分自身の 学びと参加者への知識の伝達という大きなリターンを達成できます。ただ、経験的に、期限とゴールと参加してくださる方々が具体的にイメージできる状況で 「学び」に失敗する事などそうそう有りません。一見ハイリスク・ハイリターンのように読めますが、リスクの方はほぼ 100%自分でコントロールできる種の ものです。今回は一通りのリスクをコントロールできると確信できたのでさらに、私がこういうセミナーが欲しい！と常々考えていた形式にトライしてみました。要件は以下のとおり。

⑴ワークショップ形式であること/参加者が具体的な成果物を持ち帰れること
⑵講師と参加者が一緒にリアルタイムで成果物を作り上げていくこと
⑶参加にあたって条件となる開発技術経験のハードルは極限まで下げること

開発入門講座ですので、成果物は実際に動作するアプリケーションになります。最初に講義、次にライブコーディングでそれを実演、参加者がそれにならう倣う、という形式で、通常想定される数倍のチェックポイントを設けることで講師も含めた参加者全員が同期をとりながら一歩一歩進めるような形を目指しました。実際に はこの参加者全員が同期を取る、というのはとても困難である事が容易に想像できましたので、講師の他に躓いた参加者をフォローして頂く補助の講師に数名ご参加頂きました(宮田さん、津田さん、渡辺さん、加藤さん、東さん、早川さんに感謝です。また、前述の参加者からの思いがけない質問に私が対応できないと きも、左記の方々に助け舟を出していただきました)。ワークショップ形式では必ずつまずく方がランダムに発生しますので、ここのフォローアップをどう設計 するかが非常に重要なポイントとなります。全員止めてその間に雑学や細かい部分の解説で場を繋ぐ、とりあえず先に進んで次のチェックポイントまでに補助の 講師にお任せで追いついてもらう、などの手法があります。

社内外を問わず、私が講師として何か技術やノウハウを伝えるときに特に気を付けている点として、

⑴その技術やノウハウで何が/誰が、なぜ嬉しいのか？を論理的に伝える
⑵今何をやっているのか、全体の中のどんな位置にいるのかを本当にしつこく、くどく伝える
⑶世の中の 8 割の人が同意してくれるレベルで簡略化する。2 割の嘘はいとわない

があります。特に ⑵ は有識者や他の講師の方からすれば本当にくどいので、同席されると眉をひそめられるかも知れませんが、私自身がおよそ有史以来最も鈍い「勉強会」参加者で あった経緯からして、こと「わかりやすさ」というパラメータを重視するにはこの方法が最適と信じています。当日はこの講義スタイルでトレードオフになる 「スピード」を考慮のうえ、7 時間というかなり余裕を持ったタイムテーブルを組みました。結果としては何とかギリギリある問題を解決できる最低限のアプリ ケーションまでは全員たどり着く事が出来ましたが、想定していたゴールの 60%の位置で時間切れとなってしまいました。残り 40%がとても楽しい部分だっただけに本当に悔やまれます。次回で必ず挽回したいと思います。ゴールインできて いないので明らかな失敗なのですが、散会後に数名の参加者の方から「わかりやすさ」について評価頂けたのは嬉しかったです。

【主催者として】

「勉強会」を主催するとき、テーマ以外に何が必要でしょう。ひとつは催しそのものを引っ張る役の人。これは講師であったり、発表者をアテンドする司会で あったりその形態は様々です。今回のように、主催者が講師を兼ねる例も少なくありません。ひとつは催しの告知方法。これに関してはここ 1、2 年でウェブ上 の告知、スケジュール管理ツールが充分に実用レベル、かつ気楽に使えるようになってきましたので、現在ではほとんど問題にならないかも知れません。 Twitter も自分と興味を共有している人と繋がっているという点で、とても強力な告知ツールでもあります。

テーマも決まり、講師の段取りもついていざ告知、という段になって立ちふさがる最後にして最大の壁が、会場です。最低でも 10 人程度が集まれて、プロジェ クターは必須、マイクや可能であれば参加者が自由に使える無線 LAN 環境も欲しい。となると、都内の貸し会議室では 2 時間で数万円という額になってしまい ます。企業に所属する方であれば次に考えるのが自社設備の利用ですが、これも会社によって実現性に大きな差があり、空間はあってもその入退室管理、ネット ワークの分断など主にセキュリティ上の事由により却下されてしまう場合が多々あります。例外として、既にある一定以上の知名度のある方が主催する場合、主 に外資系の企業が会場を提供してくれる例もありますが、そういった方は「勉強会のつくりかた」という段階ではないでしょうから、私たちの参考にはならないでしょう。

現状、この問題に対する解決策は 2+1 つあります。

⑴「勉強会」に理解のある企業に勤める知り合いに会場確保を依頼する
⑵地域の施設や大学、特定技術の推進団体に相談する
⑶オンラインでやってしまう

ひとつめの方法を採るにあたっては、ニフティ株式会社様による「@nifty エンジニアサポート」が現状最高のサービス形態であると言えます。主催者が自社(ニフティ)社員である必要がなく、設備面でも一切の抜かりなく、前述のものに加え動画配信用の機材まで完備しています。正直、これを最初に見たときには「やられた！」と思いました。IT 企業における最大の資産が人であるなら、自主的に「勉強会」を催したり、そこに足を運んだりする層の技術者に自社の名前を知ってもらう、よい印象を持ってもらえることを思えば、入退室管理やネット ワークの分断に掛かるたかだかの投資など、年間を通して人材獲得に掛けているコストと比べても無視できるレベルではないでしょうか。さらに、夜間や土日、祝日の会議室はいわば「遊んでいる土地」であり、管理費用や土地代だけを無為に浪費しているとも言えます。これを有効活用しない手はなく、むしろここまで 思い至ればやらないほうが不思議です。大きな会議室をお持ちの企業の特に人事や広報のみなさまには、是非一度ご検討を頂ければと思います。

ふたつめは催しものを安価にあげるのに古典的かつスタンダードな方法です。地域の公民館や区民ホールなど、専業のサービスに比べると費用面で大きなアドバ ンテージがあります。しかし、手続きの煩雑さや詳細な制約事項(使えるコンセントの口の数など)が、特に機器を多く用いる IT 系の勉強会には不向きな面も 否めません。もし伝手があれば、大学や推進団体に所属する方に相談して、場所を提供してもらうほうが設備面や制約事項の面で有利でしょう。また、いささか 裏技的ですが、”同じような催し物を開催した事のある人に聞いてみる”という手もあります。案外、こちらのほうが先方ものってくれてスムーズに話が進むかもしれません。

最後の方法を +1 とした理由は、まだ前述のふたつの方法ほど確立された手法ではないから、です。そもそも場所を必要とせず、設備や参加者全員が各個に準備、そしてブロード バンド環境さえあれば誰でもいつでも主催、参加できるこの方法が確立できれば、今後の「勉強会」手法の主流になり得る可能性を秘めているのですが、この手 法は致命的な欠点もはらんでいます。それは参加者同士の交流が薄れてしまうことです。「勉強会」は受講やディスカッション、ワークショップを経て自分の技 術力を向上させられる事が本義ですが、散会後の歓談の時間や、懇親会での交流によって新たな人的ネットワークが形成される点も見逃せないメリットです。この点を上手く解決できるようなインフラまたはやり方の模索をはじめています。

【さいごに】

以上、Android 開発入門や社内/社外勉強会のほんの少しの主催/講師経験と、数多の参加者経験を踏まえて、私なりの「勉強会のつくりかた」をご案内 いたしました。いまさら私に言われるまでもなく、凄まじい速さで変化する環境に対応するため、IT 技術者は常に学び続ける事を強いられる職業です。いえ、どんな職業でも学び続ける事は必要なのでしょうが、こと IT 技術者においてはその必要性が顕著でわかり易いように感じます。しかし、本来学びとは発見であるなら、発見をし続けることができる職業であるとも言えるでしょう。せっかく、こんなに面白い職業に就いているのですから、ふとした疑問や発見を独り占め せずにみんなで共有することで、その喜びを何倍にも広げてみてはいかがでしょうか。

プロフィール

松木 晋祐（マツキ シンスケ）

株式会社 ACCESS フロントエンド事業部 主任

NPO 法人ソフトウェアテスト技術振興協会（ASTER）会員

JaSST Tokyo 実行委員

Android テスト部 副部長

組み込みブラウザのテスト担当、マネジメントを経て本社研究開発部門の品質保証セクションの再構築及びそのフレームワーク開発を担当。現在はプロジェクトマネジメント職に従事。

ソフトウェアを楽しく素早く創るための最重要技術としてソフトウェアテストを学ぶ。

3. SQuBOK®

■ 「SQuBOK の利用法：参照スタイルから進化スタイルへの提案（その１）」

株式会社 NTT データ MSE
ソリューションサービス事業部
コンサルティンググループ 堀 明広

「SQuBOK ユーザー会 世話人」の堀です。

「SQuBOK ユーザー会」は，以下を目的に，2009 年に設立されました。

・先人が切り開き，培ってきた品質技術を伝承し，それらを有効利用して，更には発展させるため，SQuBOK 活用の事例を共有する。また，SQuBOK をより密に活用する方策・方法を模索し，共有する。

・ソフトウェア開発に関係する者にとって，SQuBOK がより価値ある知識体系に進化し続けることに，SQuBOK のユーザー自身も参画し，これを実現する。

従来の「SQuBOK ユーザー会」の活動のメインは，メーリングリストを通じての議論でいました。

今回のこの記事では、「SQuBOK ユーザー会」で今後取り組んでいきたいと，世話人が考えているとを書いています。

【まず、最初に】
皆さんは、SQuBOK をどのように使っているでしょうか。

「SQuBOK を傍らに置き、何か不明な点が出てきたら SQuBOK を取り出して調べ物をする」というように、多くの方は SQuBOK をソフトウェア品質に関する"辞書"として使っていることと思います。この使い方は非常に正しいと思います。
今や internet で、ありとあらゆることが検索出来てしまいます。情報はまさに「無尽蔵」であり、そこから必要な情報だけを抜き取ることが難しくなってきています。SQuBOK は、ソフトウェア品質に関する重要な事柄の、そのエッセンスをギュッと凝縮してまとめられたものです。　何か調べ物をする際に、まずは「SQuBOK 先生」に問うてみたら、大枠の情報が簡単に得られるものと思います。

【ソフトウェア品質に関して、皆さんはどのように勉強していますか？】

SQuBOK は「ソフトウェア品質に関する重要な事柄の、そのエッセンスをギュッと凝縮してまとめられたもの」です。（この定義は私が勝手に作ったものです）
しかし当然のことながら、SQuBOK でソフトウェア品質の何から何まで、全てが得られるわけではありません。

話はここで本筋から一旦離れ、私の経験談に移ります。
私は現在、ソフトウェア品質に関する専門職にいますが、元々は組み込み系プログラマーでした。
私はプログラマー時代からテストが好きで、ボーリス・バイザーの「ソフトウェアテスト技法」を自腹

で買って読んでいました。しかし当時は、ソフトウェアテストに関する本はこれくらいしか知りませんでした。
その後、ソフトウェア品質を専門にした職種に就き、最初にやらせていただいた仕事は、ソフトウェア品質システムの開発・維持・管理でした。ソフトウェア品質を専門としていながらも、その当時にしていた勉強と言えば、ISO9001 や CMM に書かれている内容を理解したり、それに関連した情報を少しばかり仕入れる、こんなことくらいしか、やっていませんでした。

今思うと当時は、分野がごく限られた範囲内でしかなく、しかも勉強の量も全く足りていなかったと思います。

【とある小さなきっかけで、ソフトウェア品質の世界は広大であることを知った】

もう 10 年くらい前の話ですが、当時私は、CMM を自社の品質システムに取り込む仕事をしていました。自分なりに CMM の読み込みはしていましたが、知識・理解を深めるために、CMM そのものを一通りレクチャーする研修に参加しました。
その研修は私にとって、非常に新鮮に映りました。だって、その研修では CMM の説明は少ししかされなかったのですから。
その研修の中心は、ソフトウェア品質・プロジェクト管理に関する説明であり、それぞれの内容を CMM に紐付けてレクチャーする、というスタイルが取られていました。　参考文献もたくさん紹介していただきました。
その研修を受講するまでは、私はソフトウェアエンジニアリングでどんな分野があり、どんな本があるのか、ほとんど知らない状態でいました。その研修をきっ かけに、私はプロジェクト管理、テスト、レビュー、モチベーション等々、ソフトウェア品質管理に関連する本を、片っ端から貪るように読んでいきました。　仕事のために、勉強のために本を読んでいる、という感覚は、全くありませんでした。
自分は今まで何も知らないでいたことを恥じつつも、先人が切り開いて きた道を辿っていくことは、楽しくもあり、とにかく無我夢中でした。
色々と勉強していくうちに、一言でソフトウェア品質管理と言っても、為すべきことは非常に多岐に渡っていること、品質管理の仕事は、実に創造的な活動であることを知りました。

そんな取り組みを何年か続けたのですが、ソフトウェアエンジニアリングの全体がどうなっているか、ぼんやりとした輪郭が見えているようで見えていないようで、私の中ではまだ漠然としたままでいました。

そんな状態の時に、SQuBOK が出版されてきました。

【SQuBOK で整理しやすくなった】

SQuBOK は、よく設計された書籍だと思います。
SQuBOK の目次と言うべき「樹形図」により、ソフトウェア品質に関する知見が、体系的に、読みやすく、検索しやすく整理されていると思います。

私にとって、SQuBOK は辞書的な位置づけにもあります。
SQuBOK はその他にも、自分が今までどんなことをしてきて、まだ何が分かっていないのかを整理し、

これから先は何に取り組むべきなのかを考える、”道 しるべ”のような存在でいます。そういった意味で、まさに「ソフトウェア品質知識体系ガイド」というネーミングは言い得て妙であると思います。

特にソフトウェア品質技術者は、SQuBOK の隅から隅まで一通り目を通し、何処に何が書かれているかは、ちゃんと把握しておきましょう。 何か必要が生じた時には直ぐに SQuBOK で調べられるよう、備えになります。

【SQuBOK だけでは何にもならない】

SQuBOK はソフトウェア品質に関するエッセンスが書かれたものであり、ソフトウェア品質に関する知見へのガイドです。 当該項目に関する書籍や論文、規格などが丁寧に整理されています。

SQuBOK で紹介されているこれらのエッセンスをくみ取り、更に咀嚼するためには、SQuBOK で紹介されている文献等に当たる必要があるのです。

そういった意味で、SQuBOK は「ソフトウェア品質知識体系ガイド」と銘打たれてはいますが、この本を読み進めば、どこかに自動的に連れて行ってくれる類のガイドではありません。
これらの情報を使って、自分の仕事にどう活かすことが出来るか、それは読み手にかかっているのです。
何でもそうですが、目的意識をちゃんと持っていなければ、道具は使いこなせないものです。

【ソフトウェア品質に関する知見は、常に進化している】

ソフトウェア品質に関する知見は、常に進化してきています。
世にある様々な研究発表会、シンポジウム等では、新たな創意工夫、実践事例が送り出され、共有されています。 新しい文献も、多々出版されてきています。 internet 上では、SNS や ML、BLOG 等で、活発に意見交換されています。
SQuBOK にまとめられている内容は、日常的に拡大・進化してきているのです。

【SQuBOK ユーザー会でやっていきたいこと】

SQuBOK には、今まで先人が切り開いてきた知見が体系的にまとめられています。
SQuBOK ユーザー会を発足させた時には、これら知見を蓄えた SQuBOK の活用の仕方を共有する、ということを目指していました。
つまり、今までの SQuBOK ユーザー会は、色々な知見が書かれている SQuBOK をどう利用するか、という観点で捉えていました。

上述しているように、ソフトウェア品質に関する知見、事例は常に進化しています。
これを、SQuBOK ユーザー会で積極的に扱っていきたいと考えています。

新たに出された知見を整理し、体系付けるのに、SQuBOK を軸として利用する。
勉強会等で出された意見等を、その場限りでなく、一個の体系に記録し、蓄えていく。

それらを使って、更に議論を深めていき、新たな知見を産み出していく。

これから提案したいのは、今までと目線を変えて、SQuBOK という体系を利用して、その内容を如何に充実させていくか、その充実化させる過程で議論をしていこうよ、というものです。

これらを行うには、internet 上で何らかのインフラが必要と考えています。
どんなインフラか、その運用方法はどうか。　これらを SQuBOK ユーザー会で議論していきたいと思っています。

次回の記事では、これらの活動結果がどうなったか、報告したいと思います。

プロフィール
堀　明広（ほり　あきひろ）
株式会社 NTT データ MSE
ソリューションサービス事業部　コンサルティンググループ
日科技連 SQiP シンポジウム委員会　副委員長
日科技連 SQuBOK ユーザー会　世話人
ソフトウェアエンジニアリングで好きな分野は、メトリクス。中でも見積りは特に重要であると確信している。もうひとつ好きなのはテスト。テストは破壊的であるのと同時に、創造的な魅力に溢れていると思っている。
得意技はバグ分析。バグ情報には，品質を占う情報と，改善の道へ繋がるヒントがたくさん散りばめられている。

4. トピックス

■ 特別企画 SQiP 講演会「今求められる、中堅ソフトウェア技術者の人材育成」ルポ

財団法人　日本科学技術連盟　SQiP 事務局

2011 年 4 月 21 日（木）に（財）日本科学技術連盟・本部ビルにおいて、特別企画として、SQiP 講演会「今求められる、中堅ソフトウェア技術者の人材育成」が開催されました。

本講演会は、クラウドコンピューティングや急速なグローバル化により、システム開発における大きな変革期にある中で、これを乗り切るために、優れたリー ダーシップと、変革の実務を支える中堅技術者の変革スキルの必要性を中堅技術者育成の問題意識をお持ちの教育担当様向けにお伝えすることをねらいとして実 施されました。

日科技連では、ソフトウェアにおける品質・生産性のコア技術として、テストからソフトウェア開発を再考する「Inverted Study」に基づく、分析し考える「技術者育成」が不可欠と考え、2010 年から「モダンソフトウェアテストアカデミー：プロフェッショナルコース」を開設しました。その人材育成の必要性を、今一度紐解くために、求められる中堅ソフトウェア技術者の人材育成とその研修プログラムの考え方についての講演と、そして、研修の成果紹介がありました。

当初、3 月 15 日に開催予定でしたが、震災の影響で約 1 ヶ月延期されました。それにもかかわらず、約 70 名の方々に参加いただき、盛況に行われました。

プログラム

【日 時：2011 年 4 月 21 日（木）13:30～16:50】

【会 場：千駄ヶ谷本部　1 号館 3 階講堂】

時間	内容
13:30～13:40	開催の挨拶
13:40～15:10	【講演】 「今求められる、中堅ソフトウェア技術者の人材育成～次世代を担うエンジニア～」 (有)デバッグ工学研究所　代表　松尾谷　徹氏
15:10～15:20	休憩
15:20～15:40	【SQiP の紹介】 (財)日本科学技術連盟が目指すソフトウェア品質向上への取り組み (財)日本科学技術連盟　SQiP 事務局
15:40～16:40	【2010 年度「モダンソフトウェアテストアカデミー(MTA):プロフェッショナルコース」セミナー修了生による成果の紹介】 （発表 20 分＋質疑 5 分×2 件　＋　解説 10 分） (1)「課題抽出のための業務可視化手法の提案」 (株)インテック　技術部　藤井　彩乃氏 (2)「機能単体テスト （仕様ベースのブラックボックステスト)の効率化」 伊藤忠テクノソリューションズ(株)　開発技術推進部 標準化・品質向上推進課　山本　徹氏
16:40～16:50	終了挨拶
16:50～17:30	相談コーナー（希望者対象）

１．【講演】「今求められる、中堅ソフトウェア技術者の人材育成～次世代を担うエンジニア～」

（有）デバッグ工学研究所　代表　松尾谷　徹氏

デバッグ工学研究所代表の松尾谷徹氏から以下のアジェンダに基づいて講演が行われました。

① 人材育成に関する論点
② 中堅技術者の現状スキル
③ 経験偏重スキルからの脱却
④ 今後、求められるスキルとは
⑤ 中堅技術者育成：今期の試み
⑥ 論点のまとめ

ソフトウェア技術者がどのようなスキルを習得しているか、また、そこにある問題点は何か、その解決の糸口として求められる技術者のスキルや育成方法はどうあるべきか、ということが重要な論点であると強調されました。要旨は以下の通りです。

現在のソフトウェア開発技術は 2 つに分かれており、1 つはスタッフが持つ知識的スキル（スタッフ系スキル）である。もう 1 つは現場の実務的スキル（現場系スキル）である。
スタッフは管理技術が中心である。固有技術についても専門知識を習得しているが、それは紙面上の知識というべきものであり、応用力がない。そのため、現場 からは役に立たないことを押し付けてくるととられることが多い。スタッフ系スキルは主に学習で習得することができるが、なかなか実務に結びつかないことが 多く、下手をすると事業貢献の低下につながる恐れがある。
一方、現場は自分の経験に基づく固有技術を保有するが、理論に裏打ちされたものではなく、仕事が変わると全く役に立たなくなる。つまり、現場系スキルは主 に経験で習得することができるが、理論と結びつかない特定実務を対象とするものが多いため、応用力は高めることができない。
現在は、二極化したスキルを持つ人材構成になっている。その考えられる原因は 2 つある。
1 つは現場側（ライン側）における問題として、5 ゲン主義の「原理」「原則」が欠如し、狭い「現場、現物、現実」になったしまったということと、体系化された OJT/OJD で習得したものではなくなってしまい、職場での育成力が死滅状態になってしまったことである。自分の経験した狭い製品固有技術が主になり、特定の経験に依存したスキルであるために、システム環境が変わると、応用力が低くなってしまうのである。
もう 1 つは、技術面で現場支援する立場のスタッフ側における問題として、5 ゲン主義　の　「原理」「原則」が先行し、「現場、現物、現実」から逃避してし まったことである。1990 年以降、スタッフ部門が誕生して、品質保証（ISO9000）、生産技術（メトリックス、見える化）、プロセス改善、 ITSS など、管理技術の習得・導入が行われた。しかし、固有技術に対する先輩のスキル不足から、その OJT はそもそも存在しなかった。ここでもやはり育 成できる人材が不足していたといえる。
今後、現場側では経験偏重のスキルから脱却し、知識と実践の橋渡しとなる人材育成が必要になる。

自分の経験偏重から脱却するプログラムとしては、
①実務の問題として、その全体像を知ること、
②その問題を構造的に捉え分析すること、
③先人や他者の知恵/成功からプラクティス（技法）を選択すること、
④メンバーとの課題共有し、説明できる実務技術のリーダーシップを持つこと、
が重要である。
昨年度のモダンソフトウェアテストアカデミー（MTA）は、 現場の問題を分析して課題及びその解決策をモデル化し、さらにそのモデルを具現化して現実モデルを構築するアプローチで課題解決策を立案した。これは、 1990 年代から欧米で実践されてきたモデリングによるアプローチの実践であったが、実際には参加者による自分の課題のモデリングは簡単には進まなかっ た。
今年度の MTA は、過去の成功プラクティスを示し、受講生がそのパターンと自分の課題を照らし合わせて、自分の課題にもっとも類似した解決策を選ぶこ とができるようにする。実務により近いところにある技法等の選択を行って、解決策を組み立てることで、より早く課題解決への道を見つけることができるよう にする。
また、中堅技術者の現実的な育成について意見交換や情報共有する研究会を立ち上げることを考えている。

以上のように、ソフトウェア技術者が、職場経験に頼ったスキルや、実務と遊離した知識しか持たないという問題に対して、育成の場をつくることが一番重要で あると同時に、組織力がビジネスの優劣を決定する中で、応用力の人脈を社内・社外問わず必要であることを聴衆に伝え、講演を締めくくられました。

２.【2010 年度「モダンソフトウェアテストアカデミー（MTA）」セミナー修了生による成果の紹介】

（１）課題抽出のための業務可視化手法の提供

（株）インテック 技術部　藤井彩乃氏

インテックの藤井彩乃氏から、昨年、モダンソフトウェアテストアカデミーに 参加され、その成果を紹介いただきました。藤井氏は、開発技術部の仕事はスタッフとして、開発部門と情報を共有し、要望を聞き、それに対して、支援、研修 などの協力をし、その中で、開発部門の抱える課題を解決すること、将来を見据えて先見的な施策を展開することを業務としています。

これまでの業務において、開発部門の課題を聞いてその解決策を提案・提供してきましたが、効果検証ができないため、解決策の効果を把握できず、開発部門の 真の課題が本当に把握できているのかという疑問が生まれました。このため、納得感のある効果的な施策の提案のために何をすればよいのかを、この MTA の課 題として取り組まれました。

MTA の講義の中で、DFD が業務プロセスの明確化にも役に立つことが分かり、それを適用して職場の課題解明に適用しようと決め、自身と同じ部門のメン バーがツールを開発部へ普及させようとしたのですが、なかなかうまくいかず悩んでいたので、その背景をもとに DFD を使って分析されました。

その分析をするまでは、ツール普及が進まないのは開発部門が意欲に欠けているが原因と考えていたとのことでしたが、分析結果から、ツール使用のノウハウが 特定部門に偏り、そのノウハウが共有できていないために普及が滞っていることが認識できたそうです。また、ノウハウの普及にスタッフが関係していないこと が問題と分かったため、今後、何をやるべきかが明らかとなったそうです。ヒアリングの被回答者からも、DFD の理解しやすさが評価され、今の課題（まず取 り組まなければいけない課題）を掘り下げて考えられるようになり、成果物が何かのインプットになるような活動が必要であることが明確になったとのことです。

業務の可視化による効果として、説明者が自分のやろうとしていることを他の作業との関係で整理・評価し直すことができるとともに、他者が疑問提示、アドバ イスをしやすくなることが挙げられます。したがって、今後、可視化手法として、DFD と他の方法を比較評価しながら支援作業のツールとして活用していくとのことでした。

（２）機能単体テスト（仕様ベースのブラックボックステスト）の効率化

伊藤忠テクノソリューションズ（株） 開発技術推進部 標準化・品質向上推進課 山本 徹氏

続いて、同様に昨年、モダンソフトウェアテストアカデミーに参加されました伊藤忠テクノソリューションズの山本徹氏から、その成果を紹介されました。

山本氏が携わる、自社の開発プロジェクトでは、自社と委託先による協業により複数のテストを段階的に実施しています。「機能単体テスト」（仕様ベースのブ ラックボックステスト）では、これまで、機能単体テストはチェックリストによる手作業ベースのテストが中心となっていました。

しかし、顧客のシステム化業務機能の高度化・複雑化に伴って、テストケース数が増大の一途をたどり、委託先にて期間内にテストを終了することができなく なってきました。それにより、自社に受入後も「機能単体テスト」のバグが頻発し、テスト工程に手戻りが発生するケースが多く見受けられたそうです。

問題を考察すると、機能の組合せテストケース設計は開発担当者の経験則にゆだねられており、効果的な組合せ法が研究されていないことにあることが分かって きました。この解決策として、組合せの合理的方法が世の中にあることを MTA の講義で知り、さらにその利用法を習得しました。

組合せ方法には All-pair と HAYST 法があり、それぞれのメリット、デメリットを検討して All-pair を採用しました。具体的には、フリーソ フトツール Jenny を使って機能の組合せを作成し、さらにそれをチェックリストに展開するツールを開発しました。これによって、外部委託する際にも効率 的なテストを依頼することができるようになったそうです。今後、禁則に関する指定方法の明確化、All-pair 利用箇所の標準化などを実施するとのこと です。

その定性的な効果としては、以下の 3 つが挙げられました。

①組合せテストの標準化による属人性の排除、

②自社/パートナーとの合意形成の確立、

③自社への受入基準の明確化

また、定量的な効果としては、最小限のテストケース数で効率的に機能単体テストのバグを摘出できたそうです。

成果紹介後、2 つほど質問が出ました。

Q1：テストケース数を All-pair で削減した結果、実際のテストケース数が足りなくなることはなかったのか？

A1：試行の結果ではそのようなことはない。これから本格展開して確認する。

Q2：MTA の成果は何か？

A2：講師と自分の課題を共有し、解決策を検討できたこと。

３．相談コーナー（希望者対象）

成果の紹介の発表が終了した後、希望者の方を対象に、相談コーナーを設けて、以下のような活発な議論が行われました。

Q1：問題というのは、ひとつひとつの現象ではなく、全体的な方向をとらえて対策を考えていく必要があるという話があったが、それは賛成である。その際、それをどのように解決していくのか、具体的にはどうするのか。

A1：問題は作り込んだところまで遡る必要があるが、それが行えていないために根本的なフィードバックができないであろう。テスト技法を習得するというのは、テスト方法を考えることであり、前提とする設計を考えることにつながる。テストと設計を結びつけるところに解がある。つまり、複合設計、構造設計といったものが、なぜ生まれたのか、何を解決するために生まれたのか、その点が説明されてこなかった。
そのようなテストや設計の技術に遡って
問題の根源を明らかにすることで、解決の方向を見つけることができる。

Q2：技術者が会社にいなくなってきた。プログラミングを外部に発注しているため、プログラミングの経験を持たずに、仕様設計する技術者が多くなっている。
そのような場合にどうすればよいのか、解があるのか。

A2：社内の技術者は構造設計等の知識を持ったうえで、外部発注時に制約条件を提示することが必要である。某企業では、機能が働かない無則を含めて同値分割することをやっている。
動いてはいけない条件で機能が動作しないかのテストを実践している。
品質保証、開発技術などが原理原則を習得して、発注先にも守らせることが大切である。
それを省いて、中身の分からないプログラムの保守をしなければならなくなるのは、非常によくない状態である。

Q3：モダンソフトウェアテストアカデミープロフェッショナルコースでトップアスリートを作ることも必要であるが、それを展開することも必要である。技術面はわかったが、展開にはそれ以外の要素も必要である。その展開はどのようにしているのか。

A3：分析時に展開も含めて検討している。技術者は関係性を作るのが上手とはいえないので、それに対する関係性作りも本コースのカリキュラムに入っている。また、コース開催中は、参加者の上司への状況連絡を随時行っており、成果発表会には上司にも出席してもらい、講師を含めた三者で改善を考えることを推進している。

Q4：テストをする前に、レビューや設計方法をチェックしたほうが効果的ではないのか。

A4：プログラムの構造なども重要であり、それも本コースのカリキュラムに含まれている。

Q5：危機感をどうもたせるか。

A5：社外へ出て刺激を受けることも大事である。

Q6：原理原則が必要であることを現場にどのように教えていくか、気づかせるか。

A6：原理原則を理解していないと、問題が変わると適用できない。例えば、本コースの演習では、考えなければはまってしまう落とし穴を体験するようにしている。

このように、相談コーナーに参加されたみなさん、中堅ソフトウェア技術者をプロフェッショナルへ育成する方法を悩まれている様子でありました。どんなに職 場経験を積んでも、それだけではプロフェッショナルへ育成することができないのは明白であり、エンジニアリングの知識と応用力だけではなく、ビジネスセン スや対人スキルを含めプロフェッショナルとして自立していくことが大事なターニングポイントであります。しかし、多くの職場ではターニングポイントとなる 機会に恵まれることがなく、成長の機会を失っているように思えますが、この機会は、職場という狭い井戸の中から、広い世界に飛躍することも大事ありま す。そのために、日科技連が提供する、モダンソフトウェアテストアカデミープロフェッショナルコースをぜひ活用し、プロフェッショナルが不足している検証、テストの分野を対象としたプロフェッショナル育成に役立てていただきたいです。

5. 健康

■ ～いきいき働くために、ストレスと上手に付き合おう～

株式会社プラネット・コンサルティング
代表取締役　根岸　勢津子

読者のみなさんは、心の健康と聞いてどんなことを想像しますか？そもそも心というのは臓器として存在しないだけに、なにかつかみどころがないような気がし ますね。私たちは、ただロボットのように働いて自分や家族を養っているわけではなく、その時々や、場合によって様々な気分や感情を抱えながら働いているのです。

◆まじめな A さんがうつ病を発症するまで

ある事例をご紹介しましょう。A さんは 27 歳の独身男性。今年 5 年目のプログラマです。この仕事が大好きで、やりがいをもって働いていました。両親は地方 で健在、A さんは都内のマンションに一人暮らしです。責任感が強く、まじめに仕事に取り組むので、勤務先でも評価の高い人でした。先月まで 3 年がかりの大きな開発案件に区切りがつき、今月からは違うチームで新規案件を担当することになりました。勤務先の場所も変わり、とても早く家を出なければなりません。

チームに入ってみると、そこにいたのはとても厳しいマネジャーと、物静かに仕事をする人たちでした。以前はプロジェクトリーダーの声掛けで、飲み会なども あったのですが、今度の職場では飲み会どころか普段の会話もほとんどありません。A さんは新しい仕事でわからない部分を先輩に教えてもらいたかったのです が、とても聞きづらい雰囲気で、自分で調べるしかありませんでした。調べ物に手間取って毎日終電ごろまで仕事場にいるような日が続きました。何事もきちん としないと気が済まない A さんは、毎日遅く帰宅して、急いでご飯やお風呂を済ませて布団に入るのですが、ここ 1 週間ほどよく眠れません。深夜 3 時ごろまで 寝つけず、朝が来ても全く疲れが取れていないまま出勤する日が続きました。食事も砂をかむようで味がせず、特に午前中がつらくてたまりません。遅刻しなが ら働いていたのですが、ある日 A さんは、どうしても電車に乗っていられなくなり、途中下車したホームで 1 時間近くベンチに座っていました。それでもその日 はお昼頃出勤して、なんとか仕事場で過ごしたのですが・・・。とうとう、ある朝、鉛を呑んだような気分で起きられなくなり、会社に電話を入れて休ませてほしいと伝えました。そして自分はだめな人間で、もう仕事はできないと思う、と話したそうです。

驚いたマネジャーが本社に連絡を取り、産業医の勧めで A さんは精神科を受診することになりました。診断はうつ病でした。その後、A さんは半年間の休職ののち、ずいぶん回復しましたが、医師には同じ職場に戻るのは無理と言われ、比較的負担の軽い仕事から始めることになりました。

◆心の病は手遅れになる場合が多い

いかがでしょうか、典型的なうつ病のケースをご案内いたしましたが、心の病とは目に見えにくいため、手遅れになることが多いのです。風邪であれば、鼻水が 出たり、微熱があったりすれば、体験的に「風邪かな？」とわかるのですが、うつ病をはじめとした心の病は、どういうものが前兆なのか、という知識がないために発見がおくれ、それに加えて治療に関する情報（医療機関や治療内容）が乏しいためになんだか敷居が高く、つい専門家に見せるのが遅れてしまった、というケースも見受けられます。

そして、もう一つの大きな理由が、心の病に対する偏見です。日本は世界の中でも精神病に対する偏見が強く、精神科にかかっていることを会社に知られたら首になってしまう、とか、変なうわさを流される、といったネガティブな気持ちが働き、なかなか精神科を受診するまでに至らない、ということがあります。そのようなことを防ぐためには、病気に関する正しい知識を持ち、予防や治療において正しい行動がとれるよう、みなさんも積極的に勉強していただきたいと思います。

◆ストレスと体調の関係

病気そのものを知る前に、うつ病の原因の一つと言われているストレスについて考えてみましょう。ストレスとは、日常のいたるところに存在し、ある程度のス トレスは人のやる気を高めるといわれています。しかし、過度のストレスを受けると、だれでも体調不良や感情面での乱れなどが表れるものです。ホルモンや自 律神経の調子が狂い、血圧の上昇、肩こり、食欲不振、気分がふさぐ、生理不順、といったような様々な症状が現れてきます。気分や感情の面でいえば、怒りっぽくなったり、悲しくなったり、無気力になったり、みなさんも多かれ少なかれ体験があるのではないでしょうか。かといって、全てにおいてストレスの原因そのものを全くなくすことは困難です。

ではどうしたらいいのでしょうか。それは物事の捉え方をプラス思考にすることです。自分を必要以上に過小評価したり、うまくいかないことばかりを想像する のはやめましょう。物の見方は常にふた通りあります。コップに半分だけジュースが入っていたら、あなたはどう考えますか？「もう半分しかない」「あと半分 もある」。また、新しいことに取り組んだり、新しい人と知り合ったりするとき、「きっとうまくいく」「うまくいくはずがない」。楽天主義と悲観主義がある ならば、楽天的にものをとらえるほうが精神衛生上よいに決まっていますね。このように、ストレスそのものを取り除くことはできなくても、ストレスの衝撃を 和らげたりすることは可能なのです。

危険なのは、自分のストレスに気づかぬまま無理を続けることです。人によってはストレスに気づきにくい体質の人もいます。ときにはセルフチェックテストを受けるなどして、自分の心の調子に敏感になりましょう。

◆楽天主義になるには、普段の生活リズムも大切

読者の皆さんは意外に思うかもしれませんが、明るくものを考えられるようになるためには、健康的な生活をすることが第一歩と言われています。美味しいと感じるものを、楽しい人たちと一緒に食べ、一生懸命働き、疲れてよく眠る。そして大切なのが休日のリフレッシュです。休日にも仕事のことで頭がいっぱいで は、脳が休む間もありません。休日には、仕事と全く関係ない趣味をしたり、人と会ったりして、心身をリラックスさせてあげましょう。

また適度な運動も、脳には良い刺激となります。読者の皆さんはデスクワークが多いでしょうから、1 時間机に向ったら、伸びをしたり、首をぐるぐる回すな ど、座ったままできるストレッチ体操などを心がけてください。そしてランチタイムには、できれば 15 分間程度の散歩をするとよいでしょう。外の景色を眺め て、脳をリフレッシュさせるのです。そしてまたよい気分で午後の仕事に取り組むようにしましょう。皆さんの周りにも、自分のストレスに気づかずため込んでしまっている人がいるかもしれません。そんな時には声を掛け合って、みんなでストレス解消しましょう。

◆うつ病に関する知識

それでは、いよいよ病気に関する知識を学びましょう。うつ病とは、いったいどこが悪くなる病気でしょうか。心の病といっても、心という臓器はありません。そうです、うつ病は脳の機能障害なのです。脳は神経伝達物質という電気信号で体の動きや気分・感情を制御していますが、その機能が正常に働かなくなる病気 です。また、ストレスがたまりすぎると脳の血流が低下することも最近の研究でわかってきています。気の弱い人がかかる病気とか、もともと怠けものがなる、なんてとんでもない。立派な政治家や経営者、そして芸術家などにも少なくないのです。簡単にいえば脳に疲労がたまった状態で、その人のストレス処理能力を 超えたときに、「もう駄目だ！」と脳が信号を出すようなものです。どんなに精神力が強く体が丈夫な人でも、やりがいのない仕事を長時間やらされたり、年がら年中気を使う立場に置かれたりしたら、体か心が悲鳴を上げるでしょう。

また、大きな環境の変化も病気の引き金になります。仕事上の変化でいえば、転職、転勤、昇進、降格、会社の合併など、私生活でいえば、結婚、離婚、家族と の死別、引っ越し、出産、そんなものが引き金になることもあります。だれもがかかるリスクのある病気、それがうつ病なのです。

次にうつ病の前兆ですが、寝れない（睡眠障害）、食べられない（食欲不振）、何にも興味がわかない、集中力が低下する、疲れやすい、悲しい気分、ふさいだ 気分・・・こういう症状が複数あり、それが 10 日間以上続くようであれば、うつ病の疑いがあります。躊躇せず専門家に診てもらいましょう。早期発見なら外来（通院）だけで治せます。みなさんもご自身のことだけでなく、周囲の人のこれらの前兆にも気をつけましょう。異変に気づいたら、すぐに受診を勧めてあげましょう。

◆何科で診てもらうのか、治療方法は？

うつ病は精神科で診る病気です。どうしても抵抗があるようならば、市町村の保健所や精神保健センターなどで相談してみるとよいでしょう。治療方法ですが、 現代医療では投薬と休養がメインとなります。カウンセリングは、医師の指導のもとに受けるのがよく、インターネットなどで調べたカウンセラーなどのもと に、自分の判断で通うのはあまり感心しません。

また、うつ病の特徴として、非常に再発率が高い病気といわれています。再発しないためには、正しいリハビリと周囲の協力が欠かせません。治療とリハビリは 素人では無理です。職場に復帰し、その後徐々に調子が戻り、完全に元の状態に戻るまでが治療です。かってに薬をやめたり、医師のすすめるリハビリ・プログ ラムに従わなかったりすると、容易に再発してしまいます。また、周囲の人も本人を焦らせることなく、温かく見守ってあげたいものです。

◆明るく暮らして、うつ病から身を守ろう

現代生活はストレスが多いものですから、原因そのものを消滅させるわけにはいきませんが、すでに書いてきたとおり、規則正しい生活を送り、悲観的にものを 考えない、これがうつ病予防の全てです。自分は楽しく毎日を送れているかどうか、健康的な生活ができているかどうか、これを機会に振り返ってみるのもよい のではないでしょうか。読者のみなさんが、健康で楽しく働かれることを願っています。

次回は、部下や後輩をもつ方に向けて、日常の注意事項などに触れていきたいと思います。

プロフィール
根岸 勢津子　（ねぎし　せつこ）
株式会社プラネット・コンサルティング 代表取締役
企業や団体の中で『人の役に立つためには』を常に考え、外資系、海運会社、IT 企業などで秘書の経験を積む。その後、大手損保代理店に転職したが、法人・団体を守るためには保険販売のみならず、リスクマネジメントの考え方が必要と感じ、アドバイザーの経験を積む。近年、産業界にヒューマンエラーによる不祥事が続発したことを受け、企業に対するメンタルヘルスケアの体制構築に関するアドバイスに注力して事業を進め、2006 年に法人化。メンタルヘルス対策に取り組む企業からの様々な相談に応じ、コンサルティングを行う。