

EM WEST

VOL.2 JULY 2010

THE ROADS
OF THE TEST
RUN TO
WEST
JAPAN

テストの道は西日本に通ず

[特集]

テスト&レビューを
“ガチ”で考える



◎特集

テスト&レビューを “ガチ”で考える

「ソフトウェアレビューの価値」 森崎修司……………04

「Wモデル～ソフトウェアテストによるプロセスイノベーション～」

西 康晴……………06

「JaSST Kansaiへおこしやす」 新美崇宏……………08

「アジャイルにおけるテスト開発プロセスの考え方」

ほそたに(や)……………09

西日本コミュニティ巡り

「中四国にも勉強会が増えてきた！」 横山隆司……………12

「北陸のコレ系コミュニティ事情」 大谷和紀……………14



mamezou

株式会社豆蔵

〒163-0434 東京都新宿区西新宿2-1-1 新宿三井ビル34F TEL 03-5339-2114

URL <http://www.mamezou.com/> MAIL sales@mamezou.com

ソフトウェアテストが抱える課題を
解決するぞう

お任せ
ください



テストイング
プロセス改善

テストイング技術、
ツール導入支援

テストイング教育

テストイング入門、基礎、実践講座

JSTQB 講座

品質保証入門講座

レビュー実践講座

品質分析評価技法講座

◎連載

ほそたに(あ)直伝!

ITエンジニアのためのアロマ講座 [第3回]

ほそたに(あ)16

現場での要求開発のはじめかた [第1回]

「ふりかえれば要求開発」

前川哲次17

本間直人の連載 [第2回]

「山男はなぜ山に登るのか？」

本間直人18

東京のIT革命児の聖地!! 鳥一代のサムゲタンが
関西でも!! 喰える!!

その1 楽天市場から 鳥一代 で 

<http://www.rakuten.co.jp/toriitidai/>

モバイルはコチラ!



その2 メール、FAXでもお申込み受付中!(カード不可)

メールアドレス: samugetan-tori0926@toho-e.net

FAXの場合、申込用紙はホームページからお取り寄せ下さい。

URL: <http://www.toriitidai.com/>



その2の方は、
送料無料
サービス実施中!!

テスト & レビューを “ガチ”で考える

「ソフトウェアレビューの価値」

奈良先端科学技術大学院大学
情報科学研究科
森崎修司
MORISAKI Shuji

ソフトウェアレビューがその価値を発揮するにはさまざまな条件が満たされている必要があります。

本稿では

1. そもそも実施の効果があるか
2. 適切に実施されたか?

の2点から見てみたいと思います。両方ともレビューによる早期発見、早期対策による効果が得られるか?というレビューの原理、原則に沿った条件です。

1. そもそも実施の効果があるか?

■実施の効果は「コンテキスト」次第

ソフトウェアは社会のすみずみに行きわたっており、用途、求められる品質、投入できるコスト、社会的意義等、プロジェクトに応じて多岐にわたります。本稿ではあるソフトウェアを取り巻くそれらの環境や状況をコンテキスト(文脈)と呼びます。レビューに限りませんが、技法、技術、プロセスに効果があるのはコンテキストとの整合性が保たれている場合のみです。他組織、他社で効果的だったアプローチ、技法、技術が自組織や自社で必ずしも成功しない理由の1つは開発活動のコンテキストが異なるからと言えます。

■コンテキストの例

具体的なコンテキストの例をコードレビューを題材にして紹介します。コードレビューに時間をかけるサービスAの開発とコードレビューには時間をかけないシステムBの開発を想像してください。システムBの実装に携わるエンジニアが「サービスAではきちんとコー

ドレビューやってるらしいわー。うらやましい。うちも本来やるべきなのにマネージャ連中がわかってないし。ホンマ。頼むで」という発言を耳にすることは多いと思います。私自身にも同じような経験があります。

システムBのソースコードもサービスAにない綿密にレビューすべきでしょうか?その判断材料になるのがサービスAとシステムBの開発のコンテキストです。では、サービスAとシステムBの開発のコンテキストを右の囲みのように仮定して、実施の是非を考えてみます。

囲みで記述したコンテキストのもとでは、システムBのコードレビューの実施は、コスト対効果の面で難しそうです(個人的にはどんなコードもレビューしたいのですが)。このコンテキストでは、設計レビューや単体テストに注力するほうが効果が高いと言えます。

なお、文献[2]の大規模調査の結果(特に指示しない場合、コードレビューの指摘の75%が保守、拡張に関する指摘)を前提にしています。もちろん「機能追加部分に限定して、潜在的不具合のみを指摘する」等の適切な制約を与えれば実施の効果を得ることはできます。

この例を通じて紹介したかったことは、コードレビュー自体に普遍的な価値が存在するのではなく、サービスA

サービスAの開発のコンテキスト

サービスAのソースコードは10年間にわたって保守、拡張することが決まっている。また、これまでの実績値の蓄積と分析から、可読性や拡張性の低い部分をコードレビューで指摘することにより、約10年間分の拡張をスムーズにし、開発担当者が入替わるときの知識移転コストも小さくできることが十分に期待できる。また、コア部分についてはテストコードを書いてクラス、モジュールレベルのテストの自動化によるメリットが実績値から見込まれている。

システムBの開発のコンテキスト

システムBは受託開発によるものであり、既に長年にわたって場当たりのな拡張が続けられてきている。今回の開発はそこに機能追加をする。また、その次のリリースでは大規模な再設計が決まっており、ソースコードの多くは新規に書き直される。

の置かれたコンテキストにおいてコードレビューの価値が発生することです。また、システムBにおいてもコンテキストがかわればコードレビューの価値もかわる可能性があります。

2. 適切に実施されたか?

では、次に実施の価値があるとして、適切に実施されたかどうかを見ていきます。

■レビューの価値

レビューには様々な(潜在的な)価値があり、次の3つに大別できます。こ

れらを事前に想定、設定せずにレビューを実施すると期待するレビューの効果が得られないことがあります。一度のレビューで複数の項目を同時に目的として設定することもできます。

a. 潜在的不具合の発見

ソフトウェアテストに先立って問題点や欠陥を発見する。誤った記述、曖昧な記述、一貫性のない記述等、放置すると不具合になり得るものも含む。

b. 保守性、拡張性の向上

次バージョン以降の開発において拡張、保守上、問題になりそうな箇所を検出する。

c. 担当者どうしの情報共有、知識共有

お互いに依存しあう部分を担当する開発者、品質保証に関わるメンバの間で情報共有することで勘違い、考慮漏れ、実装漏れを防ぐ。新しいメンバにこれまでの状況を移転するねらいや教育の目的もある。

■レビューの効果をj得るための前提

前述a、b、cで効果を得るための前提A、B、Cは次のとおりです。図1はこれらの前提が満たされていない例を示しています。

A. 早期発見によって修正コスト低減が

見込まれる不具合の指摘

潜在的な不具合を発見する場合、テストで発見された場合と比較して、修正コストや修正確認コストが小さいことがレビューの価値が得られることの前提です。図中では見逃した場合の修正コストとして0.2人時の指摘の割合が大きいのですが、これではレビューの効果は得られにくくなります。

B. 機能拡張、変更予定のイメージを

伴った拡張性、保守性向上の指摘

将来の拡張、保守時の改変コストよりも指摘、修正のコストが小さいこと

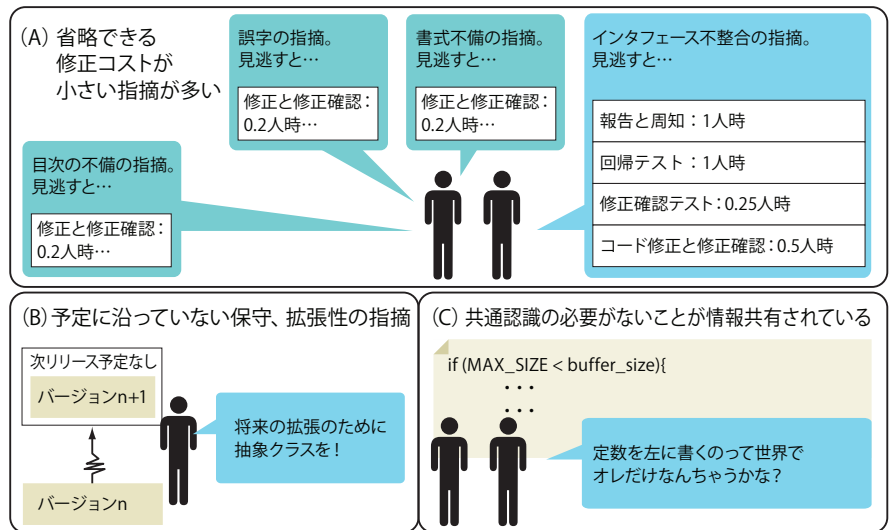


図1 レビューが適切に実施されていない例

が効果を得られる前提です。Aの潜在的な不具合よりもコストの推測が難しくなりますが、拡張、変更について方針がある場合には、非常に効果が高くなります。拡張、変更の予定がない、または不明な部分からは効果が得られにくくなります。図では次バージョンのリリースの予定のないソースコードをリファクタリングして抽象クラスを追加しようとしています。

C. 担当者間で認識の一致がずれている項目の指摘

インタフェースの一貫性や共有リソースの確保方法等、チームメンバの境界やメンバ間で共通認識を持っておくべきことや一貫性を保つ必要のある事項の確認と合意ができれば期待する効果が得られます。

まとめ

レビューの効果が得られるための代表的な条件を2つ紹介しました。2つが揃わないと効果的なレビューは実現できません。レビューが形骸化しやすい理由の1つは、レビューの効果が実感できないからです。2つの条件を再度検

討されてみてはいかがでしょうか。

1つめの条件、コンテキストと価値の関係は、コードレビューやソフトウェアレビューに限定されるものではなく、開発技法や開発プロセスにも同じことがあてはまります。コンテキストについてはエンピリカルソフトウェア工学という分野で詳細に議論されています。さらなる情報として参考文献[1]をおすすめします。

Profile プロフィール



奈良先端科学技術大学院大学
 情報科学研究科
森崎修司
 MORISAKI Shuji

奈良先端科学技術大学院大学で工学博士を取得後、情報通信企業でソフトウェア開発、開発管理、レビューに従事。現在、同大学院に奉職。ソフトウェアレビュー、ソフトウェア計測、メトリクスを研究の柱とする。研究成果を業界に還元していくことをミッションの1つと位置付け、ソフトウェアレビュー研究の国際連携ワーキンググループの主導、多数の企業との共同研究、文部科学省「次世代IT基盤構築のための研究開発：StagEプロジェクト」に従事。
 メール：smrs@is.aist-nara.ac.jp
 ブログ：http://blogs.itmedia.co.jp/morisaki/

■参考文献

- [1] V. Basili: Empirical Software Engineering: Where are we headed? (2005)
<http://www.cs.umd.edu/~basili/presentations/2005%20ICSE%20Symposium%20Talk.pdf>
- [2] M. V. Mantyla and C. Lassenius: What Types of Defects Are Really Discovered in Code Reviews? IEEE Transaction on Software Engineering Vol. 35, no. 3, pp. 430-448(2009)
<http://blogs.itmedia.co.jp/morisaki/2009/08/evolvability-f.html>で一部を紹介しています。

「Wモデル～ソフトウェアテストによるプロセスイノベーション～」

電気通信大学
西 康晴
NISHI Yasuharu

はじめまして。にしです。この記事では、Wモデルと呼ばれるプロセスモデルを簡単に紹介します。別にEM「W」ESTだからWモデル、というわけではありません。たまたまです。たまたまですよ。本当にたまたまです。だから、たまたまですってばっ！

さて皆さんはプロセスと聞いて何を思い浮かべますか？作業の順序でしょうか。開発のマニュアルでしょうか。書かなくてはいけないドキュメントでしょうか。マイルストーンの集まりでしょうか。ともあれ、プロセスによって自分たちの技術が向上したり、プロセスが自分たちの強みになると考えたことなど、ほとんどないのではないのでしょうか。

Wモデルは、プロセスによって自分たちの技術が向上していくモデルであり、プロセスを自分たちの強みにできる道具の一つです。

あれれ、それはアジャイルじゃないかとお思いの読者もいらっしゃるかもしれません。アジャイルは、Wモデルとはまた別の道具ですね。アジャイルに詳しい読者は、共通点や相違点を嗅ぎ分けてみてくださいね。

■イノベーションの2つの形

プロセスを強みにするには、何らかのイノベーションがあると嬉しいですね。「プロセスイノベーション」。ちょっと格好良いかも。

さてイノベーションと聞くと、普通は何か新しい技術を思い浮かべるかもしれません。確かにそれは、イノベーションの一つです。

しかし、イノベーションにはもう一つの形があります。日々の仕事のカイゼンを積み重ね、徹底し、常識という制約を外して極端に行うことによって生まれるイノベーションです。優れた組織が生み出すプロセスイノベーションには、この形が多いと感じています。

プロセスイノベーションを引き起こした組織に秘訣を尋ねると、「秘訣なんてありません。ただ当たり前のことを当たり前に、愚直に実践しているだけです。そうしないと負けてしまいますから。」という答えが返ってきます。

それに、プロセスというのはその組織の成り立ちや文化を強く反映しています。もし全く新しいプロセスイノベーションを導入しようとすると、オーバーヘッドも大きくなりますし、現場に強く反発されるでしょう。

プロセスイノベーションで重要なのは、危機感と連続性です。現状のプロセスではダメだ、とまでは思わなくても、何かこう、もったいないな、という感覚を持つことが重要です。そしてイノベーションを起こすための一步一步をなるべく小さくできれば、現場が負担を感じずにイノベーションに近づいていくことができます。

■Wモデルの2つの哲学： フィードバックと他工程配慮

イノベーションには哲学が必要です。ある哲学を徹底的に、極端に実現することでカイゼンがイノベーションに変化するからです。

哲学が異なると、同じようなプラクティスを実施している組織でも、イノベーションの方向性が異なります。例えば、管理・統制・文書化・指標を重視する哲学と、助け合い・ノウハウ・

実感を重視する哲学とでは、徹底し極端に実現した組織がずいぶん異なることは容易に想像がつくと思います。

Wモデルでは、フィードバックと他工程配慮という2つの哲学が必要です。

“Errare humanum est”(過つは人の常)という言葉に示されているように、人間は間違いを完全になくすことはできません。しかし、間違えた後になるべく早く気づいて直すことはできますし、一度間違えたところであればあらかじめ注意しておくことはできます。これがフィードバックという哲学です。

またフィードバックを機能させるためには、他の人に協力してもらうことが重要です。そうした思いやりの心が、他工程配慮という哲学です。後工程、他のチーム、お客様、委託先が仕事をしやすいように配慮して仕事をしたり情報を渡すことを意味します。

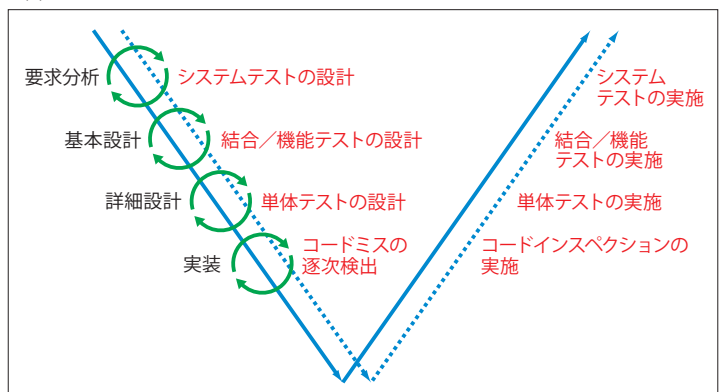
■Wモデルのコンセプト： テスト設計の前倒し

皆さんの現場では、テストを実行しなくてもテストを設計するだけでバグが見つかることはありませんか？それなら、わざわざ開発の最後にまとめてテストケースを設計してバグを見つけなくても、もっと早く見つけられるはずですね。何ともったいない。

このもったいない状況を減らすためには、テストの設計をテストの実行と分離して、開発の上流に前倒ししてやればよいのです。これがWモデルのコンセプトです(図1)。何となくWの文字に見える気がしませんか。ほ～ら見えてきた見えてきた。

Wモデルでは、要求分析を行ったらすぐに、もしくは同時に、もしくはそ

■図1 Wモデルのコンセプト



の前に、要求ベースのテストを設計します。アーキテクチャ設計、詳細設計、コーディングも同様です。すると、テスト設計を行うだけで見つけられるバグがほとんど修正された状態で、言い換えるとかなり品質が高まった状態で、テスト実行に進むことができます。テストファーストやTDDと似ている気がしませんか。

WモデルはAndreas Spillnerが最初に提案しました。日本では、ハードウェアの開発で後工程を前倒しするコンカレントエンジニアリングというコンセプトなどを基に、Spillnerとは独立にこのアイデアに到達した人が複数いたようです。もしかしたら、彼が想定していたよりも進んだ使い方を議論しているかもしれませんね。プロセスイノベーションは日本人のお家芸ですから。

■Wモデルのメリット(1)

Wモデルのメリットのうち一番単純なのは、作業順序を入れ替えて上流でテストを設計すると細かく考えることができ、抜けや矛盾を減らせるというものです。

しかし、行き当たりばつりにテストを実行していたり、コピペと微修正でテストケースを増殖させていると、テスト設計時にバグが見つからないため、テストを前倒ししても大した効果は得られません。また開発上流で膨大なテストケースを記述すると、仕様や設計の変更時に膨大なテストケースの書き直し作業が必要となり、工数超過を起してしまうデメリットも発生してしまいます。

このメリットは、あくまでWモデルの入り口に過ぎないと肝に銘じておきましょう。

■Wモデルのメリット(2)

2つめのメリットは、ソフトウェアをスッパリ作れるという点です。開発上流で組み合わせテストを設計すると、仕様や設計、コードがゴチャゴチャしている部分によってテストケース数が実行不可能なほど爆発的に増加してしまうことが把握できます。すなわち、組み合わせテストケース数を実行可能な程度に減少させるには、仕様や設計、

コードをすっきり作り直してもらう必要が生じるのです。

ここで注意してほしいのは、開発上流でテストケースを詳細に記述する必要はないという点です。テスト設計によって開発エンジニアの思考を補完するのが、テストエンジニアの本当の使命なのです。ちょうどよい粒度でテスト設計をストップすることが重要です。

■Wモデルのメリット(3)

3つめのメリットは、開発エンジニアが気づいていない観点をテストエンジニアが示唆できるという点です。例えば制御系の開発では、状態遷移に関する設計検討が抜けてしまい、当然仕様書にも書かれません。しかし腕の良いテストエンジニアは、仕様書に書かれていない観点をバシバシ思いつき、かつ俯瞰的に整理して網羅していくのです。

開発エンジニアはあらゆる観点から検討しているようで、採用した開発のやり方に固有の観点が抜けてしまいます。逆にテストエンジニアは過去の不具合の分析などにより、抜けやすい観点を知っている場合があります。開発エンジニアは開発に必要な観点を絞り込むように頭を使いますが、テストエンジニアは俯瞰的・多面的に捉えようとする傾向があります。まさに補完的な関係ですね。

■Wモデルのメリット(4)

4つめのメリットは、開発エンジニアが忘れがちな、要求分析・設計・実装工程の「由来」と「行く末」を常に意識し、両者の乖離を防ぐことができるという点です。

開発エンジニアは「なぜそういう設計なのか(由来)」や「こういう設計変更を行うとシステム全体としてはどういう振る舞いをするのか(行く末)」などを意外にもあまり考えずに開発している場合があるため、両者が乖離していきます。

一方、腕の良いテストエンジニアは、網羅的なテストを行うために「なぜそういう設計なのか(由来)」というテスト目的を考えたり、期待結果を導くために「システム全体としてはどういう振る舞いをするのか(行く末)」を考え

ます。そのためテスト設計を開発上流で行うと、「由来」と「行く末」の乖離に歯止めをかけられるようになります。

■Wモデルのメリット(5)

5つめのメリットは、開発組織の持っている弱点を明確にし防ぐことができるという点です。開発組織は同じようなバグを何度も作り込んでしまう弱点を持っていますし、ドメインにも弱点があります。腕の良いテストエンジニアは、よく作り込まれるバグをパターン化しておき、そこを狙ってテストを行っています。いわば、開発の弱点を狙うわけですね。

バグパターンに基づいたテスト設計を開発上流で行うと、開発エンジニアが自分たちの弱点を意識して開発するようになるため、最初からバグが入り込まない開発に近づいていきます。テストを開発上流で設計すると、開発エンジニアの成長をテストエンジニアが手助けできるわけです。

■Wモデルの効果: フィードバック時間の減少

もともとVモデルからWモデルへの進化は、フィードバック時間の減少が目的でした。すなわち、バグをいかに早く直して開発効率を高めるか、という考え方でしたね。

ではWモデルの5つのメリットを最大限に活かし、フィードバック時間を極限まで減らすと、どうなるでしょう。開発エンジニアとテストエンジニアと一緒に作業をすることで、開発エンジニアが間違えそうになるまさにその瞬間にテストエンジニアが気づき、実質的にバグが作り込まれない開発が可能になるでしょう。これがフィードバック時間ゼロの開発です。図1の上流のサイクルが極限まで小さくなるイメージですね。

さらに、フィードバック時間をマイナスにする開発をイメージしてみましょう。それは、開発エンジニアが開発するため何かを考えている時に、頭の中でもう一人の自分がテストエンジニアの役割となって、気づかない観点や弱点を警告してくれるような開発です。悟りを開いた状態と言ってもよい

かもしれません。これによって、居合い切りのように、一切のムダのない研ぎ澄まされた開発が可能になるわけです。

実はこれこそが、本当の意味の「上流からの品質の作り込み」です。決して最初から矛盾のない仕様を書くことでも、開発手順をマニュアル化することでもありません。開発で得られたノウハウを駆使して常に優れた「気づき」を行い、フィードバック時間をマイナスにすることで、最初からバグのない開発をしようという考え方なのです。

■Wモデルの目指すべき姿

Wモデルの目指すべき姿は何でしょう。それは、開発エンジニアが「最も

良いやり方で考える」ことに近づけていくプロセスです。

良い開発に近づいていくというのは、山に登るようなものだと考えています。どの方向から登ろうが、目指す頂上は同じです。Wモデルはテストという方向から山に登る道筋を示しましたが、唯一の道筋ではありません。ぜひ皆さんの組織でも、理想の頂上を目指してずっと登り続ける道筋を考えてみてください。登ることをあきらめたら、開発エンジニアとして終わりです。日本中のエンジニアが、それぞれの登り方を提案し実践し議論できるような世界が来ることを願っています。

最後に、この記事を書くにあたり大

きな示唆を頂いたソフトウェアシンポジウム2010のWG2の皆さんに深く感謝いたします。

Profile プロフィール



電気通信大学
西 康晴
NISHI Yasuharu

某社でソフトウェアテストに関するコンサルティング部門を立ち上げた後、東京・調布の電気通信大学にて研究や教育、コンサルティングを行う傍ら、ソフトウェアテストのエバンジェリストとして「現場に笑顔を」をキーワードに飛び回っている。甘いものと西村しのぶをこよなく好む愛妻家・愛娘家。

「JaSST Kansaiへおこしやす」

JaSST Kansai実行委員
新美崇宏
NIIMI Takahiro

EM WESTをお読みの皆さん、こんにちは。JaSST Kansai実行委員の新美です。この度、JaSSTの紹介記事を書かせていただくことになりました。JaSSTとは何か、その歴史を簡単に紹介し、続いて今年の「JaSST'10 Kansai」(<http://www.jasst.jp/>)の見所をお届けしようと思います。

この記事を読んで、JaSSTに興味を持っていただければ幸いです。

■JaSSTとは？

JaSSTは、ソフトウェアテストシンポジウム(Japan Symposium on Software Testing)の略称で、「ジャスト」と読みます。発音は、「ジャ」にアクセントを置いてもらうと格好良く、それっぽく聞こえます。

シンポジウム開催の大きな目的は、技術者、管理者、研究者などさまざまな分野で活躍するテスト関係者が積極的に参加し、研究の成果や経験を発表、共有、議論することにより、日本のソフトウェアテスト技術の向上に貢献することです。

つまり、いろんな人の話を聞いて議論しましょう、ということですね^^。

2003年に東京で開催されてから、現在は全国6地域で開催される全国的なシンポジウムに成長しました(写真1、2)。2003年といえば、ちょうどテスト技術に注目が集まり始めたころでした。まさにJaSST開催以降に、テスト関連書籍

の発行冊数が大幅に増加し始めた、そんな時期です。

■JaSST'10 Kansaiの見所

2010年7月23日(金)に、クリエイターズプラザ(クリエイション・コア東大阪3F:大阪府東大阪市)にて、ソフトウェアテストシンポジウム関西「JaSST'10 Kansai」が開催されます。

今年のテーマは、「負けないテスト」です。基調セッションでは、さまざまな企業から、その企業独自の取り組みをご紹介します。例年、著名人による長時間の講演を行っています。今年は3名の方よりさまざまな視点での講演をいただきます。

■負けないテスト戦略

今年のワークショップは、テーマである「負けないテスト」を実践する手助けとなるように、テスト戦略を立案する技法に触れていただけます。参加者の皆さんには、チームとなって課題に取り組んでいただけます。

ワークショップと並行して、チュートリアルセッションが用意されています。こちらは、毎年恒例となっております、テスト初級者向けのセッションです。テストに関する基礎的な知識と技術を知ることができるようになっていきます。特に、若い方にガンガンで参加いただきたいです。

そしてもちろん、今年もテクニカルセッションとして、関西で活躍されている方々を集めての研究/事例発表を行います。関西発の情報発信の場として、JaSST Kansaiを盛り上げてくれるに違いありません。

■ディープな話は情報交換会で

シンポジウムの内容だけではなく、もっとももっといろんなことを聞いてみたい!話してみたい!と思う方は多いと思います。そんな方は、ぜひ、シンポジウム終了後の情報交換会へ参加いただき、講演者や実行委員にその思いをぶつけてください。

情報交換会で、ディープな話が盛り上がり、さらにその後で飲み屋へもつれ込み、というパターンもあるようです^^。

■JaSSTへおいでやす

各地域のJaSST実行委員は、地元の企業が元気になるように、情報発信を続けていきたいと考えています。多くの方にご参加いただき、議論に加わってもらうことができれば、もっとももっと成長することができると思います。

本誌『EM WEST』を手にとったあなた、JaSSTでテスト技術について考えてみませんか?

Profile プロフィール



データプロセス
株式会社
ビジネス・イノベーションセンター主任
新美崇宏
NIIMI Takahiro

2006年より、何を勘違いしたのかJaSST Kansai実行委員として活動開始。テスト技術者の交流を広めるべく、TEF関西勉強会「てふかん」の取りまとめ役としてコミュニティでも活動中。

URL: <http://twitter.com/runiuni>



◀写真1 JaSSTの様子
▼写真2 ワークショップの様子





「アジャイルにおける テスト開発プロセスの考え方」

ほそたに (や)
HOSOTANI (YA)

■はじめに

2010年6月18日にインテックス大阪で行われたET WEST 2010において「アジャイル×テスト開発プロセスを考える」というセッションを行いました。本記事ではセッションでテーマとした「アジャイルにおけるテスト開発プロセスの考え方」を提案します。

■テスト設計における課題

皆さんは過去のプロジェクトで、リリース後に何件も障害が発生してしまった経験はないでしょうか？

リリース後の障害発生を防止するには、「すべてのケースを網羅的にテストすればよい」という意見もあると思います。この場合の「すべてのケース」とは、「仕様書に書かれていることすべて」ではなく、「仕様書に明示されているかどうかは関係なくバグが発生するすべてのケース」を意味しています。現実的に考えてバグが発生するすべてのケースをテストケースとして抽出するのは非常に困難です。よって、テ

ストでは、有限な期間、コストで実行可能、かつ実際の運用で発生しうるバグを最大限検出しなければならないという課題があります。

■テスト開発プロセスとは？

前述したテスト設計における課題に対する方策に、テスト開発プロセスの適用があります。

目標の品質を達成するためのテスト設計は、仕様書から等価的に導出できるものではなく、「分析→基本設計→詳細設計→実施」というプロセスの結果としてアウトプットされます。そしてこのプロセスを「テスト開発プロセス」と呼んでいます。テスト開発プロセスは、図1のソフトウェア開発のV字モデルにおいて左側の設計と右側のテスト実施の間のプロセスとして表現されます。

■アジャイルプロセスでテストを進化させる

目標の品質を達成するためのテスト

を開発することは容易なことではありません。図2のように開発したテストの質が低ければ目標の品質に到達しないことも考えられます。

テスト開発には、ソフトウェア開発と同等以上のスキル、創造性が必要であり、簡単にテスト開発を成功させることはできません。筆者は、アジャイルプロセスとテスト開発プロセスを組み合わせることで、段階的にテスト設計を進化させて目標の品質を達成することができるのではないかと考えています。

■「アジャイル×テスト開発プロセス」の提案

アジャイルプロセスとテスト開発プロセスを組み合わせることでテストの質を向上させるアプローチを提案します。

アジャイルプロセスは繰り返し型開発に分類されるので、通常は、イテレーションで追加した機能確認テスト、前イテレーションまでの機能に対する変更点および影響範囲のテスト等を計画

図1 テスト開発プロセス

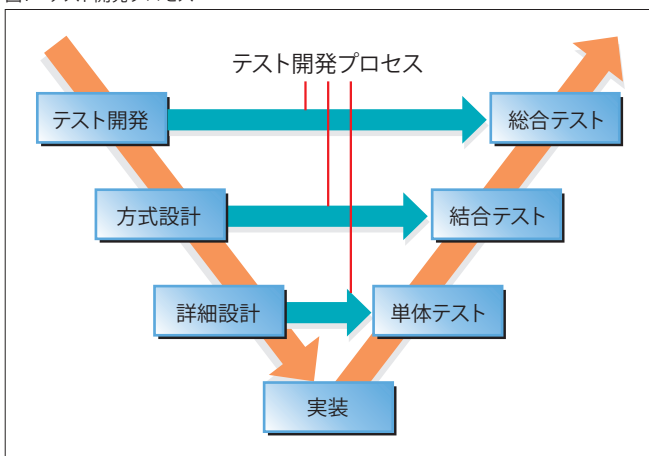
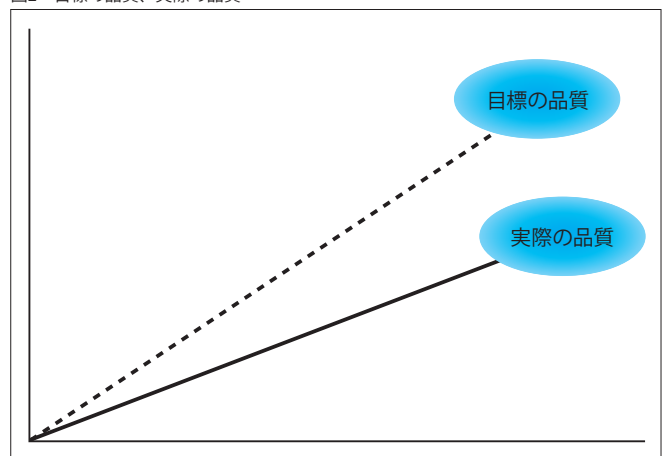


図2 目標の品質、実際の品質



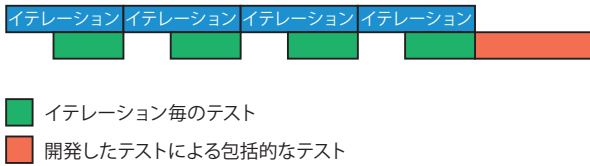


図3 アジャイル×テスト開発プロセス

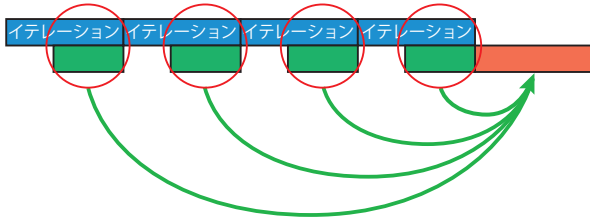


図4 各イテレーションの成果物から情報を収集

します。

今回の提案では、イテレーション毎のシステムの動作を評価するためのテスト、全イテレーション完了後に実施する、システム全体を対象とした包括的なテストを計画します(図3)。

このように包括的なテストを全イテレーションの最後に計画することで、変化する仕様に対してテスト設計を追随させることが可能となります。また、繰り返しの中で、動作するソフトウェア、設計ドキュメント、実装されたコードなどテストの質を上げるための多くの材料を得ることができます。

図4のように各イテレーションの成果物から情報を収集することにより、全イテレーション完了後に行うテストの精度を高めることができると考えています。

アジャイルプロセスでは、イテレーション毎の仕様を決定し開発を行いますが、システムの最終的な姿もリストアップすることにより、変化するゴールを追随します。代表的なアジャイルプロセスのひとつであるスクラムでは、イテレーション毎の仕様を「スプリントバックログ」、最終的なシステムに要求される仕様を「プロダクトバックログ」として管理します。同じように、テスト設計においても、イテレーション毎に実施すべきテストとシステムの最終的な姿に対するテストの両方を意識し、イテレーションの実行により得られた情報を元に進化させることによ

り、最終的なテストの質を高めることができると考えています(図5)。

■繰り返しのオーバーヘッドに関する考察

アジャイルプロセスは繰り返し型プロセスの一種であり、イテレーション毎に既存機能への変更が発生します。

繰り返し型の開発プロセスを実行する中でテストの質を向上させるというアプローチにより、繰り返しはテストにとってオーバーヘッドではなくなり、到達するのが困難な目標に到達するための有効な手段だと考えることが可能です。このため、アジャイルプロセスを導入しようとする「テストを毎回繰り返す必要がありオーバーヘッドが大きい」という意見が出る場合があります。確かに前述したテスト開発プロセスをイテレーション毎に実行した場合、変更により影響を受けてやり直しが多数発生するように思えます。テスト開発プロセスを1回だけ実施するのが最もオーバーヘッドが少ないと考える人も多いでしょう。

しかし、実際のプロジェクトを思い浮かべてください。仕様がなかなか決まらないプロジェクトや仕様変更がたびたび発生する場合には、テスト開発プロセスをオーバーヘッドなく実行することができるのでしょうか？

今回、繰り返し型の開発プロセスを実行する中でテストの質を向上させるというアプローチを提案しました。こ

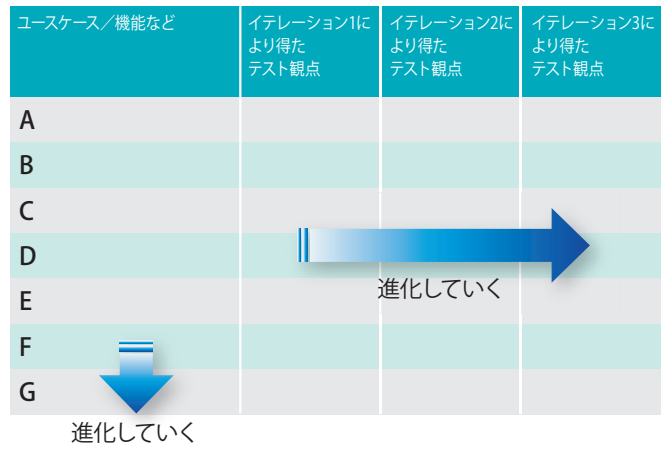


図5 テストの進化

の考え方では、繰り返しはテストにとってのオーバーヘッドではなく、到達するのが困難な目標に到達するための有効な手段となります。

■まとめ

本記事では、アジャイルプロセスにおけるテスト開発プロセスの考え方として、アジャイルプロセスにおける繰り返しが質の高いテストを開発することに貢献することを提案しました。

アジャイルプロセスとテスト開発プロセスについては、まだまだ議論が行われていない状況です。テストコミュニティ、アジャイルコミュニティがそれぞれ蓄積してきた知見を持ち寄って、議論を深めていくことが不可欠です。筆者は色々な機会を見つけてこの議論を行っていきたいと思います。皆様も是非、一緒にこのテーマを育てていただければと思います。

Profile プロフィール



ほそたに(や)
HOSOTANI (YA)

XPJUG関西代表、アジャイルプロセス協議会組込みWGリーダー。現在、アジャイルプロセスにおけるテスト、品質保証に興味を持っている。

『テストの品質』を上げるテクニックとは

ソフトウェアの品質を確保するためにはテストは必要不可欠ですが、見落としがちなのは「テストの品質」をいかに確保するのかという考え方です。テストはたいていチームで行うものですが、AさんのテストとBさんのテストを比べた時、テストの質が違ってしまっているのは、高品質なソフトウェアを安定して供給することはできません。

今回は、こうした「テストの品質」について、少し皆さんと考えてみたいと思います。

皆さんがソフトウェアテストを実施するときに、まずどんな作業を行いますか。

いきなり思いついたテストケースを書き始めますか？それとも、きちんとテストの設計仕様を考えますか？

それぞれにやり方はあると思います。

ですが、一番怖いのはテストの品質の「属人化」です。Aさんのテストは十分な観点が盛り込まれているのに、Bさんはそれに比べて、漏れている観点が存在する。これでは安心できませんよね。

そんなときに、シンプルかつ強力な手法が「テスト観点ライブラリ」という発想です。これは、テスト対象に対するテストの観点を、一覧表にまとめたものです。ベテランのテストエンジニアが持つ、「テストの観点」を、誰もが利用できる「ライブラリ」にするのです。

■テスト観点例

大項目	中項目	小項目
正常系	GUI確認	文言確認、文字切れ/文字化け
	同時動作	割り込み確認、排他処理確認
	状態遷移	バッテリー残量での動作確認
	組み合わせテスト	機能間組み合わせ、パラメーター組み合わせ
異常系	異常操作	動作中の電源OFF
	異常値入力	エラー表示確認、入力不可確認
	環境テスト	強/弱電界、ノイズ環境

これを作ることで、それ自体開発の資産になりますし、派生開発においては、大きな効率化と品質の向上が期待できます。

私たちバルテスでは、日夜こうした「テストの品質」を向上させるための取り組みを行っています。技術者間での研究・勉強会で蓄積した技術を、技術者派遣やテスト業務請負という形で、高い「テストの品質」を提供します。また、そのノウハウも、教育セミナーという形で、提供しています。ぜひ、関西の、そして日本のソフトウェア品質を皆さんとともに高めていきたいと思っています。



ソフトウェアの品質を上げるには
ソフトウェアテストサービス

- ・テストチームへテストエンジニアを増員したい
→**オンサイトテストサービス**
- ・テストを一括して任せたい
→**オフサイトテストサービス**
- ・テストの効率化を図りたい
→**テスト自動化ソリューション**
- ・プロジェクトの「強み」「弱み」を点数で見える化
→**プロジェクト品質診断サービス**

テストの品質を上げるには
ソフトウェアテストセミナー

- ・プロジェクトの品質を全体でボトムアップ
→**テストセミナー『基礎編』**
- ・テストの効率的な手法を身につけたい
→**テストセミナー『設計編』**
- ・単体テストの基本、デグレード対策を習得したい
→**テストセミナー『ホワイトボックス編』**
- ・テスト管理を確立したい
→**テストセミナー『管理編』**

バルテスウェブサイト

<http://www.valtes.co.jp>



バルテス株式会社

お問い合わせは

☎ 06-6267-6500
☎ 03-5460-9630

✉ valtes-inquiry@valtes.co.jp

大阪本社
東京本部

〒541-0052 大阪市中央区安土町 3-5-12 住友生命本町ビル 5F
〒108-0075 東京都港区港南 1-8-40 品川プレイス 8F

TEL : 06-6267-6500 FAX : 06-6267-6501
TEL : 03-5460-9630 FAX : 03-5460-9631

西日本 コミュニティ巡り

west japan community round

中四国にも勉強会が増えてきた!

横山隆司

YOKOYAMA Takashi

コミュニティが花盛りです。首都圏ではほぼ毎日、関西圏でも週1回程度はどこかで勉強会が開催されるようになってきました。「関西以西にはそんなのほとんどない!」と思われているようですが、意外にもがんばって活動しているコミュニティがたくさんあります。そこで本記事では筆者が中国、四国地方を回って自分の目で見てきた勉強会をご紹介します。

特にURLは載せませんが、Googleなどで検索していただければすぐに見つかると思います。ご興味が出てきたら、検索して足を運んでみてください。

岡山県

1. オープンラボ岡山: 毎月第3土曜日に開催されています。参加者は大体15~30名ぐらいです。ちなみに、私は「皆勤」を続けています! 勉強会の内容はさまざま、Kernel、Ruby、Robocon、セキュリティ、Java、CMS…。私もたまに、Google関係の技術を発表します! 基本的に発表時間は5分~2時間の間で自由に決められます。きっと楽しい話が聞けるのではないのでしょうか。そして、積極的に発表してお互いに

成長できたらよいのではないのでしょうか。

2. オープンセミナー@岡山: 2008年から始まって、年1回の開催で、今年は5月15日に開催されました。実は私が実行委員長を務めていました。今年はRubyのまつもとゆきひろ氏、GC本の中村成洋氏、四国GTUGの瀬戸直喜氏など、豪華な講師陣に登壇していただきました。参加者は事前予約193名、当日参加180名前後、懇親会は70名強という、中四国では最大級(?)のセミナーです。

扱うテーマは大体そのときの新しい話題が多いように感じます。2009年はOSASKの川合秀実氏に登壇していただきました。オープンセミナーで日本の最先端の技術者と並んでセッションをやってみませんか? 実行委員長が2011年から交代することになりました。私がどういう形でオープンセミナーに関わるかはまだ決めていませんが、当日皆さんのお越しをお待ちしております。ちなみに、私は「皆勤」です!

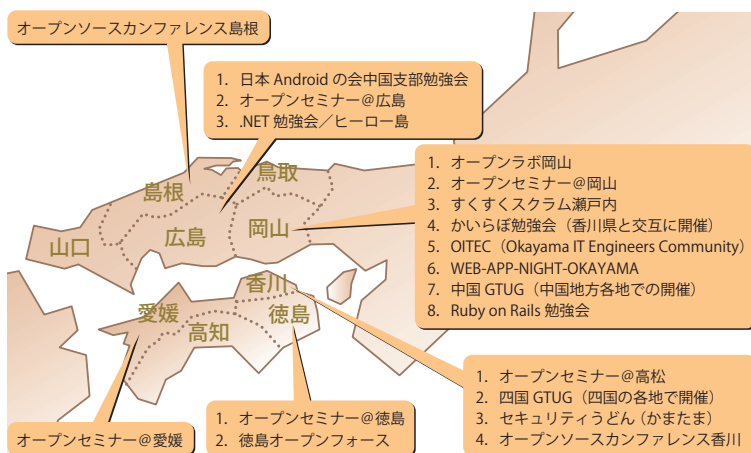
3. すくすくスクラム瀬戸内: 大体3カ月に1回程度のペースで開催されている、アジャイル開発をテーマにしたコミュニティで

す。参加者は大体20名前後で推移しているようです。2010年のオープンセミナー@岡山でも登壇された前川哲次氏が主催しています。私はまだアジャイル開発については詳しくないですが、このコミュニティでデスマーチが減るようになればいいですね。講演内容も独特で、初回で「要件定義の嘘」なんていう話題が飛び出すなど(プロジェクト全体で7~8割が失敗しているらしいです)、開発者というよりは管理者向けのコミュニティかもしれません。私が講師を務めた別のコミュニティとぶつかってしまって、「皆勤」を逃してしまいました。

4. かいらぼ勉強会(香川県と交互に開催): 3カ月に1回ぐらいのペースで開催されています。特徴は「主催者が学生2人」で、学生の参加者が多いです。取り扱う内容もセキュリティやゲームプログラムやCTF(Capture The Flag)などが取り上げられています。私は今のところ、毎回講師をしていて、Google Goでお話をさせていただいています。参加者10~15名ぐらいでまだ規模は小さいですが、学生ががんばっている姿を見るのはよいことです。学生だと思ってなめてかかると痛い目を見ますよ(笑)。今のところ「皆勤」ですが、「東京開催」を予定しているそうですので、そのうち皆勤を逃しそうです。

5. OITEC

(Okayama IT Engineers Community): 岡山で唯一(?)のMicrosoft関連技術を扱うコミュニティです。先日初参加させていただきました。参加者は大体30名程度でした。Google関係を専門でやっている私はアウェイな感じがしましたが、皆さん温かく対応をしてくれました。Microsoft MVPがぞろぞろいてレベルも高く、良いコ



コミュニティだと思います。岡山でMicrosoft関係の知識を得るにはここがいいでしょう。コミュニティメンバーのきよくら氏にはオープンラボ岡山でも発表していただいております、コミュニティ間での連携もうまく取れています。

6. WEB-APP-NIGHT-OKAYAMA: 最近できたコミュニティで、岡山では珍しい「平日開催」です。基本的には技術的な話をするというよりは雑談ベースで交流を深めるといった感じのコミュニティです。先日第1回が開催されて参加しましたが、10人以上の方が参加してくれました。岡山が熱い理由がわかった気がしました。多分みんなヒマなんですね(笑)。

7. 中国GTUG (中国地方各地での開催): つい先日活動を発表した、筆者が主催する勉強会です。Google技術に興味がある方は参加してみたいかでしょうか。岡山だけでなく、中国地方であればどこでも開催する予定で、メンバーに加入しなくても参加可能ですのでよろしくお祈りします。第1回は7月31日です (<http://atnd.org/events/5013>)。

8. Ruby on Rails勉強会: Ruby on Railsを扱う勉強会です。先日、参加してきました。発表してもよし、黙々とコードを書いてもよし、自由に質問してもよいという自由な勉強会でした。私はRuby on Railsを知らなかったのですが、黙々とコードを書いていたのですが、3時間程度でそれなりのアプリケーションが作れるのはすばらしいですね。

香川県

1. オープンセミナー@高松: オープンセミナーの歴史が一番長いです。岡山と同様に年1回の開催で、大体30~50名の参加者がいらっやいます。昨年は相馬純平氏に講演していただきました。今年は9/18(土)に開催予定と聞いています。

2. 四国GTUG (四国の各地で開催): 主催者はオープンセミナー 2010@岡山で発表していただいた瀬戸直喜氏で、Google Technology Users Groupということで、話題がすべてGoogleの技術を絡めた発表になります。私も2月にAndroidで発表をさせていただきました(そのときGolangのTシャツを頂いたので、またどこかで発表しないと...)。最近はAndroidの話題が多いですが、徳島などではGAEとかMapReduceなんかも話題に上がっているようですね。

3. セキュリティうどん (かまたま): セキュリティ関係の勉強会です。特徴としては途中でおやつタイムがあるなど、参加者にやさしい勉強会になっています。Microsoftの安納さんに登壇していただいたりしています。

4. オープンソースカンファレンス香川: 今、OSC事務局と調整していますが、2011年1~2月あたりの開催目標で進めています(四国は各県持ち回りで毎年開催を目標にしているようです)。私は直接関わっていませんが、恐らく何らかのお手伝いをする事になるとと思います。まだ情報が少ないですが、ご期待ください!

愛媛県

1. オープンセミナー@愛媛: 今年新たに発足したオープンセミナーです。岡山とか高松などと同じく年1回の開催です。どれくらいの参加者になるか楽しみです。今回はテーマを「データベース」にしているようで、いろんなデータベースで熱い議論が展開されるでしょう(本記事を読む頃にはもう終わっているかもしれませんね)。オープンソース系のデータベースに加えて、Microsoftの安納さんに登壇していただく予定となっています。

徳島県

1. オープンセミナー@徳島: 意外と熱い徳島のオープンセミナーです。大体40~50名の参加者がいらっやいます。昨年はMicrosoftの田辺氏にWindows Power Shellについてお話していただきました。今年は10~11月あたりでの開催を予定しています。

2. 徳島オープンフォース: 白衣を着たヒドラスタイルという非常に怪しい、いや独特な格好で活動している河野氏が主催している勉強会です。私はまだ参加したことがないのですが、内容はDebianとネットワークで毎月1回、交互に勉強会をやっているようです。3月にはオープンラボ岡山ともUstream通信でコラボレーションさせていただきました。オープンセミナーでも河野氏にライトニングトークをしていただきましたし、また行かないといけないなと思っています。

広島県

1. 日本Androidの会中国支部勉強会 毎月1回開催です。山口県で開催すること

もあるようです。次の開催は7月24日(土)で前は私も発表しました。これから中国GTUGとも連携しながら発展できたらいいかな?と考えていたりします。Androidはかなり熱いですね!

2. オープンセミナー@広島: 2009年にできた広島のオープンセミナーです。昨年はPostgreSQL関係の話題で固めていました。今年はどういう話題が出てくるのかまだわかりませんが、これから成長すると思いますので注目してください。

3. .NET勉強会/ヒーロー島: 私はまだ参加したことがないのですが、Microsoft関係の技術を扱っています。メンバーにはMicrosoft MVPが揃っていて、レベルの高い話題が出てくるのではないのでしょうか。

島根県

1. オープンソースカンファレンス島根: Rubyの聖地ということで(?)、毎年開催されるようになりました。オープンセミナー@岡山と日程的に近いため参加できていませんが、いつか行ってみたいです。

●まとめ

全体的に岡山が多いのは私が岡山在住だからかもしれませんが(偏りがあるのは仕方がないということで)、IT勉強会カレンダーなどを見ると、例えば広島県にはもったくさんコミュニティがあります。まだまだたくさんコミュニティがありますので興味があれば、参加してみてください。

Profile プロフィール



横山隆司
YOKOYAMA Takashi

2009、2010年
オープンセミナー
@岡山実行委員会
委員長を務める。
2年間も実行委員

長だったため、普段は「委員長」と役職で呼ばれることが多いが、実行委員長ではなく、中国GTUGのマネージャとなったので「マネージャ」と呼ぶのが正解(ちゃんと名前はありますよ)。

中国GTUGの活動はまだ始まっていないが、楽しい運営を模索しながらGoogle技術の研究や運営方針を考えたりしつつ、フリーランスのエンジニアとして活動中。

北陸のコレ系コミュニティ事情

大谷和紀

OTANI Kazunori



写真1 TDD Boot Camp北陸の様子

北陸のコレ系コミュニティや勉強会の紹介ということで、書きます。書かせていただきます。

どんなコミュニティ?何やってるの?

2008年までは勉強会らしいものはほとんど開催されなかった北陸地方。なのに2009年、なぜか突然盛り上がり、勉強会ラッシュが起こってしまいました(ちなみに北陸といえば富山、石川、福井の三県を指すことになっています)。それ以来、月2~3回、10~40人程度の規模で何かしら勉強会が開かれています。

主なものを挙げると、

- FxUG (Adobe Flex User Group)
- Hokuriku.NET (Microsoft系)
- Python.FIT
- WCAF (Web Creators Association Fukui)
- 北陸エンジニアグループ (メタ・ネタ)
- FITEA (福井情報技術者協会)
- 日本Androidの会

などなど…。

技術系、プログラミング系寄りの勉強会が多いのが特徴です。お互いにバッティングしないよう日程調整しつつ、やりたい人がそれぞれの企画して進めていく。全体としては緩いつながりで、だけどときどき濃い。そんな雰囲気です。

2009年ふりかえり、なんてのをやりました

北陸エンジニアグループの企画として、2009年12月に1年間のふりかえりイベントを開催しました(執筆時点ですでに半年以上経っているというのが…)。

ファシリテーターを務めていただいたのは、東京から来ていただいた串田幸江さん。プライベートで偶然金沢に来ていたところを、勉強会に駆けつけてくれました。同行の旦那さんを一人置き去りにして!

ふりかえりとして、7人ずつチームに分

かれて、まず1年間で参加・開催した勉強会を付箋紙に書いて模造紙に張り出し、タイムラインとしてまとめます。次に、それぞれの勉強会で感じたことをまた付箋紙に書き、貼っていく。最後に「良かったこと」「良くなかったこと」「今後チャレンジしたいこと」…いわゆるKPTをそれぞれ話しあう。こんなやり方で、コミュニティの1年をふりかえりました。

「より幅広く」「より実践的に」

そのときに出された今後のTryは「より幅広く」「より実践的に」の2点。現状、知っている人しか知らない、つながりがある人しかつながっていない状態で、もっと広く「参加したい人」を集められないか。参加しても、講演で話を聞くだけでは今ひとつ溝を埋められない。この現状はなんとかならないものか。これが打破できれば次の「何か」につながるのではないかと…。

ということで最近、金沢工業大学や石川高専、福井高専などの学生さんを巻き込んだ企画を立てたり、スマートフォンをテーマにビジネス寄りの講演会を開催する人がいたり、より幅広いイベントや勉強会が生まれています。さらに、参加者を逃がさない(!)ために書籍をプレゼントと称して「当選者は次回発表してください」ルールなんかもなく存在しています。発表者としてより深く関わる人が増えることで、より厚みが増したコミュニティになっていけばいいなあと思っています。

「より実践的に」のアプローチとして、Google HackathonやTDD Boot Camp北陸(写真1、2)など、参加型の企画も増えています。Python.FIT (Fukui, Ishikawa, Toyama)のPython勉強会)では3箇所の分散会場で開催して、ネットを通じて画面を共有しながらチャットで議論したり。それぞれのチャレンジが続いています。

今後の妄想として「仮想案件を進めよう」という案も出ています。わかっている

範囲での最良のアーキテクチャ、開発環境、開発方法論を惜しげもなく取り入れて、何かのプロジェクトを進めてみたいんじゃないかというわけです。実案件では色々なシガラミだらけで、自分たちが良いと確信していることでも採用するのは難しい。利害関係者を説得しているうちに日が沈んでしまう。それなら、仕事から離れたところで好き放題やってみようという企画です。経験も積めるし!実績も作れるし!

とまあ、こんな雰囲気です

自分で書いていて取り留めのない感じがですが、実際に取り留めのないコミュニティなので…。参加者もお仕事はWeb業界、SI業界、組み込み業界、電気屋さんなど業種は様々、仕事上の接点はほとんどなかったりするの、基本的につながりは緩いです。家庭持ちも多くて、勉強会にもなかなか顔を出せないとかの事情もあったり(って、これは私の事情)。

でも、2009年ふりかえりイベントの後、串田さんはブログで、こう書いてくださいました。

「タイムラインを拝見して驚いたのは、実にたくさんのイベントを開催されていたこと。こういう場をみなさんが渴望しているのかな、ということタイムラインから感じました。」

この“渴望”の源泉は何なのか。今後どんな情熱が引き出されるのか。コミュニティとして参加者の何かを引き出すことができるのか。参加者としてもイベントの企画者としても、これから先、とても楽しみにしています。

Profile プロフィール



大谷和紀

OTANI Kazunori

いわゆる“二次請け”業務系SEとして、要求定義から運用支援まで、ユーザ対応からプログラミングまで広く浅くやっています。77世代、子育てクラスタ、ときどきお料理ブロガー。

URL : <http://d.hatena.ne.jp/katzchang>
<http://twitter.com/katzchang>

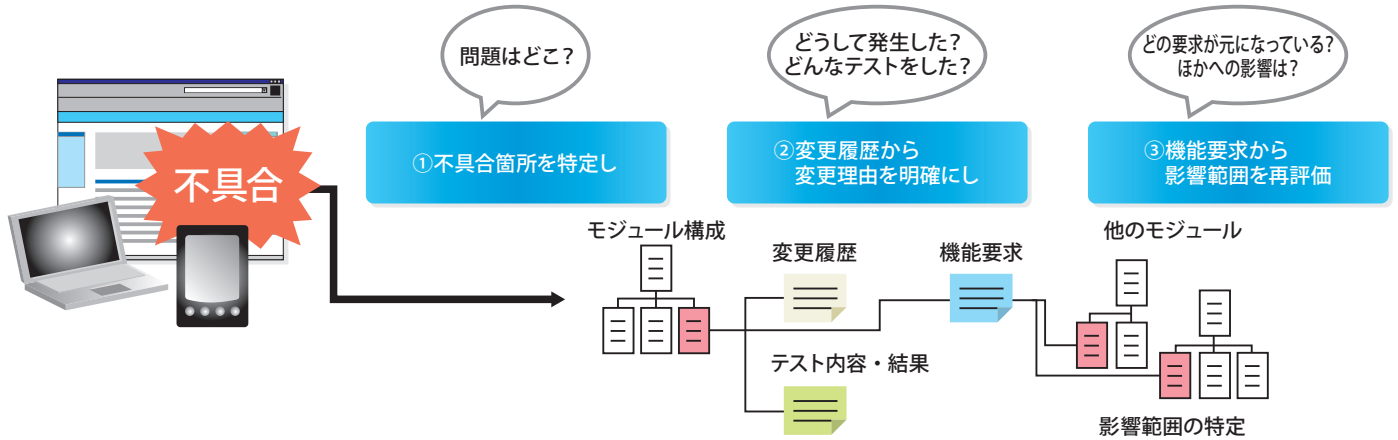


写真2 TDD Boot Camp北陸の様子

テストは Rational

ますます短納期化、複雑化するソフトウェア…

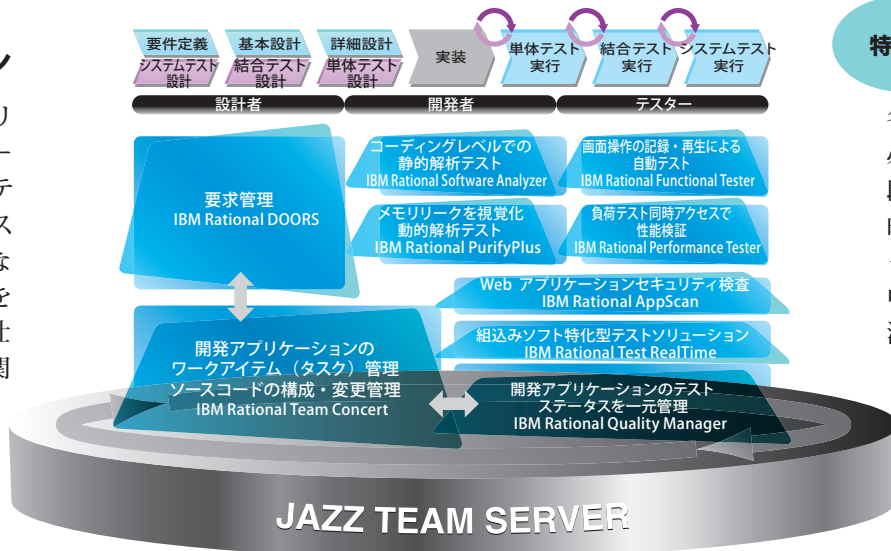
障害が発生したときに、再発防止をコミットするためにも、
原因の明確化と影響範囲の特定を、変更履歴や関連ドキュメントを参照しながら、
迅速に対処できることが必要です。



そのためには、**テストの設計、実施、結果に体系だった管理、自動化、トレーサビリティの確保、可視化が必要!**

特長1 トータルソリューション

Rationalは、開発アプリケーションのテスト・ステータスを一元管理し、単体テストから結合テスト、システムテストに至るトータルなテスト・ソリューションを提供しています。要件・仕様とテストケースの対応関係をもれなく管理することで、ソフトウェアの高い柔軟性と品質を確保します。



特長2 優れたコストパフォーマンス

各種テスト・ツールのうち必要な部分から始められ、段階的に拡張可能。効率的に品質確保を、プロジェクトの予算が限られている中、低コストで実現する方法をご提案します。

特長3 使いやすさ

Rationalのテスト・ツールは、標準的な統合開発環境であるEclipseベースのインターフェースを採用するなど、使いやすさを追求しています。習得に時間がかかることはありません。使い慣れた操作で、テストの設定、実行、分析を行えます。

Rationalはテストに必要なすべてをサポートします

詳細はWebにて。「Rationalテストソリューション」で検索してください!

日本アイ・ビー・エム株式会社 ラショナル事業部

ご相談窓口: ibm.com/software/jp/rational の「見積り依頼・ご購入相談」をクリック

電話番号: 0120-450-260 受付時間: 平日9時30分から17時30分まで (12時から13時を除く)

ほそたに(あ)直伝! ITエンジニアのための アロマ講座

第3回

ほそたに家
ほそたに(あ)
HOSOTANI (A)

みなさん、こんにちは。ほそたに(あ)です。

心身ともに元気に過ごしていらっしゃいましたか?

1回目、2回目に精油のお話をしましたが、「飲んだらあかん」ばっかやんか、というご意見をいただきましたので、今回は「飲んでええもの」の「ハーブティー」についてお話しします。

●ハーブって?

ハーブの語源はラテン語のHerbaで、草本性の植物、つまり草を意味すると言われていきます。今は一般的には花や実もハーブと呼ばれていますね。ハーブと言ってしまえばなんだか難しいと思われがちですが、そんなことはありません。

たとえばショウガのような、飲んだり食べたりして体を温める効果のあるものもハーブ。とても身近なものですよね?

●ハーブティーを煎れてみましょう

ハーブティーには大きく分けて2種類あります。と言っても、ハーブがフレッシュ(生)かドライ(乾燥)か、だけです。それぞれに良いところがあるのですが、今回は手軽に飲めるドライハーブティーをご紹介します。では、早速煎れてみましょう。

【用意するもの】

お好みのハーブ、ティースプーン(慣れたらなくてもOK)、熱湯(沸騰したてが望ましい)、ティーカップ(マグカップでもなんでも)、ティーポット(お急須でもなんでも)

【煎れ方(1人分150CCとして)】

- 1) ティーポットにティースプーン1杯程度のハーブを入れる。
- 2) お湯を注いですぐにふたをする(香りを逃さないため)。
- 3) 待つ…ハーブの種類によって時間が変わる(花・葉:3~5分、実・茎・根:8~10分)
- 4) ハーブティーの濃さが一定になるように軽くティーポットを横に振る
- 5) 召し上げ

濃い目に煎れて氷を入れたグラスに注ぎ、アイスで飲むこともできます。

【注意点】

ハーブティーは薬効成分が多く含まれているので、一度にたくさん飲み過ぎないようにしましょう。1日に5杯を限度に。また、妊娠中の方は医師の指示に従ってください。

●ハーブティーをブレンドしよう

「おいしいやん」を「おいしいやん」にするために

クセの強い、あまり単品では飲まないようなものを飲んでいませんか? 筆者はラベンダー

やローズマリーなどを単品で飲むお茶は得意ではありません。香りが強すぎないように感じるからです(ブレンドでほんの少し入っているのは好きです)。

単品で飲むなら、筆者はブラックマテがおススメです。マテ茶は、ほうじ茶のような味でとても飲みやすく、「飲むサラダ」と言われるほど成分が豊富です。

いろいろなハーブをブレンドしてもよいでしょう。

ハーブティー毎の効果を簡単に紹介します。

- 1) リラックス…カモミール、オレンジピール、ペパーミントなど
- 2) 排泄・便秘…レモングラス、ハイビスカス、ローズヒップなど
- 3) 免疫・風邪予防…エキナセア、マロウブルー、ジンジャーなど
- 4) 消化器系…カモミール、レモングラス、メリッサ、など
- 5) ホルモンバランス…ローズレッド、マリーゴールド、フェネルなど

●おススメのブレンド

- 1) ほそたに家一押し: ハイビスカス2、ローズヒップ1、レモングラス1、エキナセア1
- 2) 風邪予防!!! : ローズヒップ3、エキナセア1、ペパーミント1、カモミール1

他にほそたに家がよく飲むのは、監修の辻さんオリジナル「いたみ時間~花~」※。飲みやすいうえに、高いハーブが惜しげもなく使われています。目の疲労回復に良いと言われているマリーゴールド、のどに良いと言われているマロウブルー、あとはハイビスカスとレモングラスのブレンドです。

※いたみ時間…伊丹市にある精神障害者さんの通所施設にラベルを依頼しているので、ご愛飲くださると利用者さんのお仕事にもつながっていきます。

今回は、「正統」ハーブティーについてお話ししました。これからクーラーの時期ですから、全般的に身体を温める作用のあるハーブティーを、ぜひ飲んでほしいのです。第1回に



予防医療としてのアロマに少し触れましたが、予防医療としての植物の成分を有効に活かすにはやはり「続けること」がポイントです。で、続ける秘訣はやっぱり「おいしい・簡単」なことではないでしょうか。なので、次回は!! 「邪道」ハーブティーをお送りします!! おいしい変わった飲み方とか、実験とか…お楽しみに。

Profile プロフィール



ほそたに家
ほそたに(あ)
HOSOTANI (A)

精神保健福祉士、社会福祉士、ソーシャルアロマセラピスト。家族の立場からITエンジニアを日々観察中。ITエンジニアとパートナーのためのアロマ講座を企画中。アロマの相談があれば ayano.hosotani@gmail.comまで。

監修
植物セラピー alo-alo.
(<http://alo-alo.ocnk.net/>)

辻 朝恵

- 関係…ほそたに(あ)つながり
- 目標…冬までに遠赤外線効果のある手を持つほそたに(あ)を認定セラピストにする
- 好きな香り…月桃(げっとう)

「ふりかえれば要求開発」

前川哲次

MAEGAWA Tetsuji



1 要求開発はいつでもはじめられる

みなさんは、「要求開発を始めたいけどどうやって始めたらいいのかわからない」とか「要求開発を適用することを社内ですら承認をとればいいのかかわからない」といったことでお悩みではないでしょうか？

でも、実際には、そんなに大げさに考える必要もなく、またプロジェクトの開始前から大げさに始めなくても、要求開発をうまく取り入れて効果をあげることは可能なのです。ここではそんなテクニックの一つ、「ふりかえり要求開発」(筆者命名)をご紹介します。

2 要求開発導入のパターン

Mike Cohnは"Patterns of Agile Adoption"の中でアジャイルの導入には

- Start Small or go All In?
- Technical Practices First or Iterative First?
- Stealth Mode or a Public Display of Agility?

という3つの軸でのやり方の選択が可能であると述べています(注)。要求開発においても、似たようなことが言えます。

- 一部の組織やプロジェクトで小さく始めるか、ビッグバン式に一気に始めるか？
- 個々のプラクティスからできるものをピックアップして始めるか、全体のプロセスとして導入するか？
- ステルスマードでこっそり始めるか、おっぴらに宣言して始めるか？

もちろん、上記の組み合わせにおいては一概に「どちらを選ぶのが正しい」というものではなく、会社や組織の風土、プロジェクトの性格や状況などのコンテキストによって、何がいいのかを考えて自らの意思で選択すべきものです。

今回、この記事で筆者が紹介したいのは、この分類で言うと、それぞれ3つのペアの中でそれぞれ前者を選んだパターン、すなわち、「一部の組織やプロジェクトで、適用可能なプラクティスやツールを選んで、こっそり適用してみる」というのに近いものです。

3 要求開発を始める時の現実

実際の現場では、プロジェクトの立ち上げに先立って、要求開発に関する議論から始められることは極めて少ないと思います。むしろ、読者のみなさんがよく経験されるのは、要件定義から基本設計に入るあたりでプロジェクトが迷走し始めて、その捩入れにプロジェクトに投入されて、「さあ、どう立て直そうか？」というパターンではないでしょうか？

そんな時に、「今のままじゃダメだ。一旦プロジェクトを中断して、頭から要求開発でやり直そう！」と言い出したらどうなるでしょう？うまく上層部の理解が得られて強力なリーダーシップが発揮されたらそれも可能でしょう。しかし多くの場合、そこまで頑張ってきた現場メンバーの反発を買い、モチベーションが下がり、プロジェクトがバラバラに…という負のスパイラルに入ってしまう。

また、そのようなプロジェクトでは、ステークホルダー間、特に「経営トップ」「現場ユーザー」「開発者」といったいわゆる要求開発における「こたつモデル」を形成すべき当事者間でのコミュニケーション不全が起きているケースも多く見られます。

そこで、筆者がお勧めするのが、ユーザー企業も開発会社もひっくるめて巻き込んだ形での「ふりかえり要求開発」というやり方です。

4 ふりかえり要求開発

(再整理からの気付きと改善)

例として、システム化企画が終わり、要件定義から基本設計のあたりで、なにかモヤモヤしている状況を考えてみましょう。そのモヤモヤの原因は多くの場合、「なんとなく関係者にヒアリングしながらこのシステムの目的とかゴール、それに基づく要件の優先順位づけとかやってきたけど、本当にそれでいいのかな？一応、聞くべき人にはヒアリングしたけどこのまま実装まで進んじゃっていいのかな？」といった漠然とした不安です。

ここで役に立つのは、要求開発の考え方をを使って、合意形成のプロセスを可視化し、追跡可能性や説明可能性を担保するための「再整理」です。そしてこれは、

そこまでやってきたことの「否定」や「やり直し」ではなく、あくまでもそれを十分に生かした上での「再整理」であることがポイントなのです。

具体的にどうするかというと、システム化企画書や計画書、要件定義書など、その時点であるドキュメントを全部出してもらい、その中に要素として出てきているものを、一旦、要求分析ツリーなどの要求開発のツールに当てはめてみるのです。そしてそれを不完全であってもそのままメンバーに提示します。もしそこで欠けている要素があったり、説明のつかない部分があったりすれば、メンバーが自ら気づきを得て自発的に軌道修正していくことができます。人間は、人から指摘されたり押し付けられたりすると反発しますが、自分で気づいたことに対しては素直に対処することができるものです。

このようにして、現場メンバーとしてはモヤモヤしていたものが整理され、かつステークホルダーに対して自信を持って説明ができるようになります。また、上層部にとってもその整理された情報をもとにプロジェクトの状況や方向性を確認でき、適切な意思決定を行うことができるようになります。

5 まとめ

いかがでしょうか？「どうやって始めたらいいのかわからない」とか「社内での承認をどう取ればいいのか」とか悩んでいる場合じゃないですよね？この、「ふりかえり要求開発」で効果実感→じゃあ次は最初からやってみようという循環につなげていくことでみんなが幸せになっていく…そんな気がしませんか？

注) <http://www.mountaingoatsoftware.com/articles/38-patterns-of-agile-adoption>

Profile プロフィール



前川哲次

MAEGAWA Tetsuji

某大手Sierを経て、某ユーザー系企業の情報システム子会社に勤務。要求開発アライアンス西日本やすすくスクラム瀬戸内を立ち上げ、要求開発やアジャイル開発による価値創造を日々探求している。認定スクラムマスター。

山男はなぜ 山に登るのか？

NPO国際ファシリテーション協会理事、
NPO学習学協会理事
本間直人
HOMMA Naoto

登山家、山男山女は、なぜ山に登るのでしょうか？

古典的な答えは「そこに山があるから!!」。これは、いつも「なぜエベレストに登るのか？」と同じ質問ばかりされることに嫌気がさしたイギリスの登山家マロリーがやけくそで、適当に発した言葉とも言われていますが、記者の創作という話もあります※。いずれにしろ有名な話です。

でも、実際は、そこに山があっても登らない人はいます。「疲れる」「大変」「登ってもお金にならない」「意味ない」「お風呂に入れない」、登らない理由はいっぱいあります。

登る理由もまた多くあります。もちろん、報奨金が、名声が、という、外的動機もあるかもしれません。給料もらえるから山に荷物届けている人もいます。逆に、荷物運んでくれる人をお金払って雇ってでも、登ろうとする人もいます。頂上を極めた達成感、登れるようになった自己成長感、初めてできた稀少感、苦勞を乗り越えた仲間との協働感、内的動機付けは多くあるように思います。

その内的動機付け、山に登ると良いことがあると山人は、なぜ知っているのでしょうか？

知り合いの登山家から、「あの山に登ってよかった」との素敵な話を聴いて、憧れたのかもしれませんが。あるいは、自分自身、過去に違う山に登ってうれしかったことがあって、次は、あの山と思っていたのかもしれませんが。そうでなくて、人生で最初の山は、学校の遠足で、親に連れられいよいよで、ご褒美とか別の目的につられて、偶然たまたま、何でもいいのですが、「うれしいなんて知らなくても、登ってうれしかった初めての体験」があって新たに生み出されている可能性はありませんか？最初の一回は、「やらされ感」でも、登ってうれしいことがあれば、次は「自発的に」登る。そこに、「喜びがある」と、いわゆる強化学習、さらに、山に登る「自発的意欲」はより強くなる。そんなことはあるのではないのでしょうか？

でも、「もう一度」と思って登っていくと、あれ？立ちちはだかる障害!!「うーん、これを乗り越えるとあの喜びがきつとある、喜びを信じてをもう一度」と想像して、がんばって乗り越えると喜び倍増。今、案や障害を乗り越えてその向こうの喜びを手に入れて、立派な山男は育っていくのではないのでしょうか？もちろん、山女も。経験が増えると、喜びの種類もまた、増えていきます。大人になると味覚の種類、幅が広がるように。

仕事でも人生でも、やらない理由があっても、それでもやる理由を選んで自発的にするための動機付けの開発が可能で、重要だと私は思います。他の人との協働で、相手にその気がないと、あきらめていませんか？

いつも相手が協力的で好意的とは限りません。が、無視状態とか、敵対的とかとも限りません。その中間の状態から、可能性を見だし、協力、関係性をどう築いていくか？山登りと同じで、たとえ低い山でも仕事でも一緒にやって、喜べれば、意欲は引き出せる、意欲は育つと私は信じています。

検査検収、テストでは、テストする側、テストされる側、対立の構造になると、どちらの側でもできてあたり前みたいな構造になりがちです。ミスは、仕様の間違い、想定甘さ、コミュニケーションギャップ、プログラム上のミス、いろんなことが原因になりますが、できてあたり前。テストをされている方は、どうも、自分の能力をとにかく言われているみたいで、監視する人・される人、階層構造を思わず作ってみたい。何でこんな失敗をという自己無力感、やり直しだあという徒勞感、締め切りが、、、という焦燥感、どちらかという自発的にでなく義務で、やる雰囲気だと考えるとあります。

ペアプログラミングでプログラム作成とテストとを交互に実施すると、検査する側の対立でなく、一緒に、完成度を確認しながら高めていく、一緒に山に登っている、協働感が生まれることも期待できます。同じ仕事かもしれませんが、関わり方で登山と同様、後ろに喜びがイメージできれば、力が出ることもあるかと思えます。

※元の英語を調べると、“Why do you climb Mt. Everest?—Because it is there.”これだと、山がではなくて、エベレストがあるから、“未踏峰の世界一高い山、エベレスト”がそこにあるから挑戦を続けているという意味の気もします。達成感、自己成長感、稀少感、仲間との協働感がベースにあるように思います。

Profile プロフィール



NPO国際ファシリテーション協会理事、
NPO学習学協会理事
本間直人
HOMMA Naoto

研究開発職を経て、現職。講演活動では、コミュニケーション、ファシリテーション、コーチングなど、人と組織の活性化について、講演、研修講師として定評がある。笑いあふれる参加参加型研修が特徴。

URL : <http://www.homma.com/>

『チーム力をつくる3ステップ』

(翔泳社)

「きちんと指示しても伝わらない、動いてもらえない」「メンバーのやる気がない、部下が何を考えているかわからない」「チームの雰囲気が悪い、人が辞めるけど手が打てない」などなど、こんな悩みを持つリーダーにお勧めです!

ITエンジニアはチームで仕事をする事が多く、それだけにチーム力が低下すると、仕事の効率も品質も下がり、そして仕事が増える、といったマイナスの悪循環を作り出してしまう。良い循環をどう作るか?実際のプラクティスや身近な実例を一



緒に体験しながら、目からウロコの気づき満載です。

□本間直人の公開講座日程

8月10、11日 (2日コース、東京・渋谷、社団法人日本経営協会)
「管理者のためのチーム・ビルディング入門」

9月4日昼 (東京) XP祭り「アジャイル学園祭」

□PMのためのファシリテーション (東京、PMI日本支部)

・基礎コース (2日間コース) 9月11、12日

・応用コース (2日間コース) 8月21、22日

詳細情報、お申し込みは<http://www.homma.com/>から。

テスト駆動開発

テスト環境の仮想化と自動構築

全般的なWeb技術の
Webテスト、ロードテスト

今の Visual Studio を見てください。

Visual Studio 確かな品質



<http://www.microsoft.com/japan/visualstudio/>

評価版
ご提供中

アジャイル計画ツール

UI自動テスト

デバッグの記録、再生、巻き戻し

包括的なライフサイクル管理

●編集後記

こうして無事にVol.1に続いて怒濤の2ヵ月連続発行ができましたのも、毎度のことながら、企業スポンサー各社および個人広告出稿者のみなさま、そして記事執筆者のみなさんのおかげです。

さらに今回は、JaSST'10 Kansaiとのコラボを実現することができました。実行委員長の森崎さんをはじめ、素晴らしい原稿を寄稿して下さったみなさん、そして関係各方面との調整に奔走して下さった細谷副編集長にも改めて感謝の意を表したいと思います。

個人的な思い入れから言えば、北陸および中四国の元気なコミュニティの紹介記事が掲載できたのがとてもうれしいです。WEST って京阪神だけじゃないんだよ、と。

それから最後に、鳥一代! 確かに以前から仲間うちでは「Agilista (もしくはアジャイラー) の三大

聖地」と呼んではいたが、「IT革命児の聖地」とか自分で言うか!? (笑) そこまで言うなら、このEM WEST Vol.2を持って「EM WEST見た」と言えば高麗人参酒ロック一杯サービス! くらいのをやってほしいものだ。誰か試してみてください!

(前川 哲次)

なんとかVol.2を発行することができました。てつ。編集長から副編集長を拜命し、特集の企画をさせていただきました。無理なお願いを聞いてくださった特集の執筆者の方々、ありがとうございます。

(細谷 泰夫)

EMシリーズを一言で言うなら「コミュニケーションプースター」だと思います。企画、執筆、編集、

デザイン、校正、印刷、配布、宣伝、広告などなど至るところでコミュニケーションを生み出します。そしてそれを最も理想的に実現しているのがこのEM WESTだと思います。

いまだ紙に印刷して配るのは大変なところもあるのですが、大変だからこそコミュニケーションが生まれ、工夫が生まれます。滞りがちなEM ZEROもがんばります!

(野口 隆史)

今回あんまり関わってないばんだです^^; いや、もう、WESTはWEST編集部だけで大丈夫だろう。これが西日本のかか!むしろZERO。WEST編集部の活発な空気をEM ZEROにも取り入れていきたいと思っています。

(どこかのばんだ)

EM WEST [イーエム・ウエスト] Vol.02

2010年7月23日発行

デザイン: ミヤムラナオミ

編集長: 前川哲次

副編集長: 細谷泰夫

編集: 東秀和、前川直也

編集協力: EM ZERO編集部

発行元: 株式会社マナスリンク

〒162-0012

東京都中野区本町4-48-17-803

<http://www.manaslink.com/>

お問い合わせ先: contact@manaslink.com

印刷所: 昭栄印刷株式会社

<http://www.shoei-p.net/>

Copyright ManasLink

Printed in Japan

◎広告出稿のお願い

EM WESTでは広告を掲載して下さるクライアント様を大募集中です。企業・団体・個人は問いません。広告効果の可能性を感じていただける方がいらっしゃいましたら、お気軽にご連絡ください。西日本のエンジニアのためにも、EM WESTをどんどん発展していきたいと思っておりますので、ぜひともよろしくお願いたします。

■個人広告のお申し込み

<http://www.manaslink.com/contact/ad-personal/>

■企業・団体広告のお問い合わせ

<http://www.manaslink.com/contact/ad-company/>



瀬戸内方面の方よろしくお願いたします!

アジャイル開発
イタランゲージ
システム思考
学習する組織
プロジェクトファシリテーション

7月から松山で仕事することになりました!

twitter:kdd
<http://giantech.jp>
mailto:takeshi@giantech.jp

アジャイルコーチ
Takeshi Kakeda / 懸田 剛

ゆこ。の。こ。り。ん。広。告

I @f_yuko EM WEST

鳥一代公認チキチキ
第69代ミス参鶏湯

鳥一代の参鶏湯も
応援しています♪

お断り: この広告はネタです。

astah*でも使えます!

astah*
(旧JUDE)唯一の
公式ガイドブック

指示型のプロジェクトマネジメントになってませんか?
プロジェクトに笑顔はありますか?
活発でパワーのある
プロジェクトに変えるには……、

ファシリテーション、コーチングなど、
基本となるノウハウがこの一冊に!
技術評論社 ISBN-13: 978-4774133652

1st Album 「侍塊s」

お買い求めはこちら <http://katamaris.jp/>

各種イベントオフィシャルソング目白押し!!

¥700 Now on sale

デモを含む全8曲入り!
あなたの明日の開発の友!

1. Dear XP ~[XPJUG]XP祭り2006 オフィシャルテーマソング~
2. ぎゅっとずっと♪ ~[翔泳社]DevelopersSummit2007 オフィシャルテーマソング~
3. XP音頭 ~[XPJUG]XP祭り2007 オフィシャルテーマソング~
4. Develop ~[翔泳社]DevelopersSummit2008 オフィシャルテーマソング~
5. 約束 ~[技術評論社]EngineerMind Vol.9 タイアップ曲~
6. たまり場 ~[ManasLink]EM ZERO オフィシャルテーマソングにして欲しいソング~
7. Dear XP (English & Duet Version) *
8. 先送り (DEMO Version) *

侍塊s
(C)侍塊 http://katamaris.jp/