

WACATE マガジン Magazine Vol.45

✿ご挨拶✿

WACATE-Magazine をご覧の皆様、こんにちは。
WACATE-Magazine 編集部です。



急にあたたかくなりましたが、
皆さま体調はいかがでしょうか。

春眠暁を覚えず…
朝起きれず、昼は花粉が気になる季節になりますね。
あたたかくなると
桜が早めに咲くかも知れませんね。
今から楽しみです。

さて。突然ではございますが、
WACATE-Magazine は次号をもって
終了することとなりました。

今回が最後の WACATE レポートとなります。
とくにご覧ください！

それでは WACATE-Magazine vol. 45、
はじまるよー！

✿お品書き✿

【特集1】

WACATE2012 冬 Report!

【特集2】

JaSST' 12 Tokai Report!

【リレーコラム】

Software Test Topics

【連載】

源太郎のソフトウェアテスト占い

【あとがき】

✿WACATE Topics✿

WACATE2013 夏キックオフ！

WACATE2013 夏の日程決定！
2013/06/22(土)~23(日)

現在鋭意検討中！

WACATE2012 Winter ~三浦の熱い冬~ Report!

こんにちわっす
うえだです♪♪♪

WACATE2012 冬が終わると、すぐに年末年始になって、お正月が明けると、年度末に向けてバタバタしていて、ふと気がつく「WACATE2012 冬が終わってもう2ヶ月くらいになるのかぁー」って状況です。
参加されたみなさんはいかがだったでしょうか？
今回も非常にアツイセッションが目白押しだったと思いますっ^^
そんな WACATE2012 冬のレポートです!!!

場所はもちろん、冬場は太陽の日照時間が少ないので、イルミネーションが見れるマホロバ・マインズ三浦です！



ロビーにはクリスマスツリーも飾ってあり、まるで WACATE の5周年を盛り上げるかのようでした。

そう、今回は WACATE の最初の開催日と全く同じ 12 月 14 日~15 日という日付でもあり、**ちょうど WACATE の5周年となる、節目の日でした。**

この節目をご存じであった有志ファンによる、とても綺麗で豪華なお花が届いており、実行委員も感激していました。

お名前は存じ上げませんが、ファン一同皆様、有難うございました！
文字通り、華を添えていただきました。

感謝感激です。



そして、受付開始です。



これからの熱い二日間を前に、ロビーでもさっそく常連の皆さまが挨拶を交わすなど、ここ数年で WACATE は朝から温かい場所になっています。
これも、より熱くなるための準備運動なのかもしれませんね。
そして受付を終えると、熱い二日間が始まりました。

◆オープニングセッション 近江 久美子 (WACATE 実行委員会)

オープニングセッションは、2 日間に渡るワークショップの「注意点」や「過ごし方」の説明をします。
今回は近江先生によるオープニング。



会場内がゆるふわとした空気に包まれたところで実行委員長の挨拶が…

センターを練り歩く赤い悪m…人影。

脱皮(←)をしたそこには、金色に輝くアフロをまとった実行委員長…



スター☆やまさきの姿が！

実はこの御姿、単なる趣味ではありません。
アイスブレイクを目的としたプログラムの一環なんです。
身体をはっても参加者の方へ楽しく2日間を過ごして欲しい…
そんな覚悟が伝わる光景です。

さていよいよセッションの内容へ。
WACATE 参加者には、ネームプレートが配布されます。
近江さんは、セッションの中で、ミニミニワークとしてこんなことを言いました。



『ネームプレートの裏に、次のことを書いてみてください。このWACATEで何をしたいですか？ 何を持ち帰りたいですか？』

2日間の話題のために、自分自身の目標可視化のために、そして、2日間が終わった後の振り返りのために。短い時間の中で、より加速するための仕組みを適用させた近江さん。そして、それを見事に使いこなした参加者の方々の姿が、生き活きと記憶に残っています。

(あすか)

◆ポジションペーパーセッション 山崎 崇 (WACATE 実行委員会)



次はWACATE名物、ポジションペーパーセッション。申し込み時に記載するポジションペーパー（通称ポジペ）を使い、自己紹介や熱い想いを伝えるセッションです。限られた時間で、できる限りわかりやすく複数人に伝える。思ったよりも時間が長かった場合のアドリブなども人それぞれで、その様子を見るだけでも勉強になります。

一人の言葉に全員が真剣に向き合います。



◆ベストポジションペーパーセッション

島山 さつきさん

「かえる井の中ふりかえる～私のテスト自動化経験～」

ポジペセッションが終了したら、次は前回のベストポジションペーパー（BPP）受賞者によるセッションです。いち参加者から発表者へステップアップをしていただきたいという思いを込めて、WACATEでは前回のBPP受賞者の方に発表をしていただいています。



初めての発表に若干緊張気味の島山さん。

テスト自動化ばかりやっていたので知識不足という大きな壁に当たったとのこと。そこで出会ったのがWACATE。

WACATEではみなさん同じような悩みを抱え、次の一歩を踏み出すために参加されたという声が大変多いです。実際に先に進めた体験談は実行委員としてもとてもうれしいですね。

講演の中では、これまでやってきたテスト自動化を振り返りがされました。



結合テストの自動化
複雑なシナリオテストの自動化
外部装置の操作
複数の端末を利用した操作
帰属テストの自動化
試験書からのパラメータ自動生成
単調な仕込み作業
性能測定の自動化
再現率の低い不具合の再現
新システムへの移行

テスト自動化をかじったことがある人にはとても共感できる内容です。

#参加者の中にはガリガリ自動化を進めている方はどれくらいいたのでしょうか？

これら「なんとなく過ごしてきた日々」がTABOKなどの知識体系に触れることで

「価値に気づいた」とのこと。そして次の道に進んでいく様子が次々に語られました。

自分が今まで経験してきたことが、ソフトウェアテストの大海原ではどういう位置づけなのかを把握することで、経験に対する自信がつく。

未来へ視線を向けられる 意識の変化に気付く

自分を見直す、位置づけることは重要ですね。今一度見つめなおそうと思った方多いのでは。また、前回の初参加からわずか半年での加速っぷりには、聞いていた皆が驚かされ、刺激を受けたのではないのでしょうか。

私個人も大変考えさせられました。

畠山さん、ありがとうございました！

(さわちよ)

◆ワークショップ

中野 さやか (WACATE 実行委員会)

「テストアプローチにデータ分析を使おう」

通称姐さん、なかのさんのセッションでは、データ分析結果をどのように活用するのかを体験する機会がありました。

WACATEらしく、WACATEを対象に考えてみるセッションとなっています。

テストをする人の数だけ存在するといってもいいくらい、テストの方法は種類があります。

何のテストを、どんなふうに、どれくらいの時間をかけてやろう？

効果的に欠陥を検出するにはどうしたらよいだろう？

姐さんはそう問いかけました。

そして、どのようなテストをしていくか考えるために、データ分析の結果を見てみよう姐さんは提案します。数字や数式の抵抗感を出来るだけ排除して、まずは結果に触れてみる。

その想いをこめて、個人ワークが始まりました。



個人ワークは、様々なグラフや表から、気になった箇所を探すこと。

皆さんすぐ集中をして演習に取り組んでいました！



グループワークは、前プロジェクトのテスト実施報告書をもとに、問題点を見つけ、改善点を考えること。随所に散りばめられたヒントをもとに、問題の根底を探り、改善へとつなげていく…まるで推理小説を読んでいるかのような、そんな光景が広がっていました！



つつい議論が白熱してしまったり



会場全体を見ても、立ちあがって意見交換をしている様が見てとれます。



付箋を使ってまとめてみたり、模造紙にまとめてみたり



身を乗り出して皆で作ります。



班以外の参加者と意見交換をするために、班としての意見をまとめていきます。



実行委員も見守る中、最後の仕上げ。



各班の発表です。スライドではなく模造紙なので、皆が見える位置に集まって真剣に見ます。



「このデータはこういう意味にも取れます！」などさまざまな意見が飛び交い、互いの世界を拡げていきます。



姐さんは最後に、今回行った方法が総てではない事を強調しました。



“銀の弾丸”はない。

ここでもやはり、この言葉が脳裏にこだまします。

(あすか)

◆ゲリラセッション

そして今回2日目のモーニングセッション担当のNHN 村上 くにお氏によるゲリラセッションが！

村上氏は以前 WACATE 実行委員であったときに「線・マニックス」というデータを取り扱うセッションの講師を担当されました。

参加者のみなさんの取り組みを見て、我慢できずに「オレに一言言わせろ！」というノリで資料を作っていました。



キーワードは「メトリクスはともだち！こわくないよ！」



これできっと参加者のみなさんもデータ好きになったに違いありませんw

◆ワークショップ2

近江 久美子 (WACATE 実行委員会)

「かいてみよう CFD」



再び登場の近江さん。今度はテスト設計技法のひとつである「CFD(Cause Flow Diagram)」のセッションです。CFDとはWACATE 2012冬のクロージング講師である松尾谷 徹氏が開発した仕様記述をデシジョンテーブルに変換するための中間成果物として記載する図です。

CFD (Cause Flow Diagram) の基礎から、実際に書けるようになるまで導きます！

◇まずは、基本

CFDの特徴と生い立ちから、

- ・原因
- ・結果
- ・流れ(線)

で表される図であることを説明しました。

同値分割→CFD→デシジョンテーブルと展開していくことを伝え、順番に追っていくことになりました。

一つの例題をじっくりと掘り下げるのは、理解が進みますね！

◇復習がてら 関連が深い技法

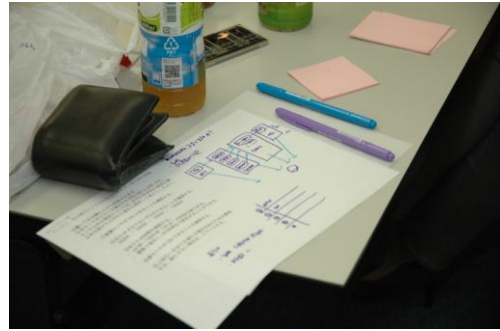
話が拵がった同値分割とデシジョンテーブルの復習です。近江先生の講義は続きます。



◇腕試し！

ここまで聞いてきた内容を踏まえて、個人ワークが始まりました。

電子レンジの設定にまつわる課題では、皆さんすらすら。



用意した課題を参加者のみなさんが図にしていきます。学びながらこのように手を動かして理解を深めていく。これがワークショップならではの醍醐味です。その後は続けて共有の時間。グループでの活動は白熱しますね！

どのような図を書いているのか、実行委員とともにイブニングセッション講演者の清水 吉男氏も見て回ります。



◇踏み込んで、+α

さらに効果的に、効率よく情報を整理するために、近江さんは講義を続けます。

- ・優先的にテストしたいものは、有効系にすること。
- ・同値クラスの状態が、原因と結果との両方に出ることもあること。

◇視野を広げて

最後に、近江さんは何故 CFD を使うのかを話します。

- ・全体を意識、俯瞰しやすいから。
- ・「今見ているものの外側」に目を向けやすくなるから。
- ・ステークホルダー同士の情報共有や説明に使いやすいから。

全体の意識や、図であることは、共有にとっても役立ちそうです！



セッションを終えた後は、松尾谷氏より参加者のみなさんへメッセージが。



講演していただくだけでなく、こういったサービスをしてくださる講師の皆さまには頭が上がりません。

◇セッション感想

CFDを書いたことがない人でも、頑張っついていくことでCFDがどのようなものを体験できる、素敵なセッションだったと思います。

私個人も設計などでCFDを使ったりすることがあるので、とても腹落ちする内容でした。

そして、すらすらとCFDを描く皆さんが、WACATEの凄さを実感させてくれました。

(あすか)

◆イブニングセッション

清水 吉男氏(システムクリエイツ)

「PFDの真価～あの遅れはどこに消えた?～」



プロセス表現技法「PFD」や仕様記述技法「USDM」、それらを組合せた開発プロセス「XDDP」の提唱者、清水氏によるセッションです。

清水様は、派生開発の分野で有名な方で、また、2009年冬のWACATEのクロージングセッションでもご登壇してくださっています。本セッションを心待ちにしていた参加者も多いことでしょう。

前半には、自分のプロセスは自分で設計しよう、というお話がありました。特に若手の頃は先輩から事細かに指示される場合も多いかもしれませんが、作業する自分自身がプロセスを設計すれば見積もりも容易になりますし、リーダーになったときにもその経験が生きてきます。そして、プロセスを設計するときに便利なツール「PFD」について説明してくださいました。PFDは成果物とプロセスの関係を示す図です。

プロセスを一度設計しても、それで終わりではありません。『計画段階に考えたプロセスが「最善」であるはずはない』のです(『』内は本セッション資料より)。プロセスを、よりよい形に変化させる必要があります。PFDでプロセスを示しておけば、このときにも役立ちます。

本セッションの終盤、健康に、そして、よい人生になるようにPFDを使ってほしいという一言がありました。参加者の皆さんにとって、仕事や人生をよりよくするツールが、また1つ増えたセッションだったのではないのでしょうか。

(近江)

一こうして、怒涛の一日目・別館の部が終了しましたが、まだまだWACATEは続きます。

一日目・本館の部としてのトップバッターは…

◆ディナーセッション



豪華なディナーに舌鼓を打ちながら、使った頭をほぐしていきます。

とりあえず一休憩!



登壇した実行委員などが率先して悪ふざけを試みたり。



セッションはこれだけではなく、参加者の意気込みの紹介や協賛団体によるプレゼントの抽選、参加者によるイベントの紹介などの情報交換などが行われます。

◆夜の分科会

そしてディナーが終了したあとは WACATE の名物となった「夜の分科会」。

各自が思い思いの議題を持って熱く語らう時間です。

◇KENさん Presents

「テス嬢 (テストのお話)」



◇国広さん Presents

「見える、見えないを身体を使って体験する」



◇水野さん Presents

「社外活動を組織とつなげよう」



◇すすきさん Presents

「バグ罫ワーストプラクティス テストエンジニア双六」



某コロケで話題のネットワークエンジニア双六！そのテストエンジニア版を作ってみたらどうなるか？

そんな動機からスタートしたテストエンジニア双六！単純に双六をプレイしたかった！のではありませんw 様々なドメインや立場の参加者で、ソフトウェアテストエンジニアのワーストプラクティス (通称ワーブラ) を共有し、そこからソフトウェアテストエンジニアの人生を楽しく考えてみる！そんな目的のもと、ミニワークショップ形式で語り合っていたのです。

出来上がった双六を見て、今後テストエンジニアを目指す人に、こんな悲しいこともあるけれど(?)、笑える楽しいことも待ってるんだ！と教育要素も盛り込んでみました。



涙や笑いのエピソードに加え、皆の知っているキーワードも満載のマス作り！夜の分科会ではまだまだほんの一部だったので、早く完成させたいですね。これからの活動に期待ですっ

大人の悪ふざけではなく、大人の真面目な分科会だったのでした♪

◇「ゲリラ分科会」



テーマだけでなく、その場のノリでお話をしたりするのも、夜の分科会の醍醐味です。

一そして、夜は更け、二日目の朝を迎えました。



◆モーニングセッション

村上 くにお氏 (NHN Japan)

「キャリアプランはじめの一歩」

今回のモーニングセッションは、かつて WACATE 実行委員だった村上くにおさんが担当。「『キャリア』について」と題したセッションを行いました。



村上さんは冒頭、資料も説明も韓国語のみでセッションを始めて参加者の度胆を抜きます。そして参加者を唖然とさせたまま、本題の解説に移りました。

まず村上さんは「エンジニアとは何か」「どのようなエンジニアになりたいか」という問いをじっくり行いました。そしてエンジニアの定義として、スキル標準や Wikipediaなどを引用しつつ、エンジニアにとって大切なこととして、「好きな技術を探そう」「世の中を変えよう」「幸せになろう」の3つを提唱。エンジニアリングやエンジニアが得られる幸せについて解説を交えつつ、説明を進めました。

次に村上さんは具体的な方向性としてキャリアについて解説。キャリアの基準として ITSS、ETSS、UISS のスキル標準や、村上さんが携わるゲーム開発でのキャリアについて紹介しました。



そしてキャリアについて一通り説明した後で、エンジニアの次のステップについて説明。エンジニアの次のステップでは、「マネージャ」「スペシャリスト」「コンサルタント」「生涯エンジニア」の4つがあり、どの職種でも ROI が強く求められると言及。より良いサービスを早く・上手く・安く実現するため、今からとにかく勉強して地力をつけておきましょうとアドバイスを行いました。

そして最後は、勉強を頑張るって悔いのないエンジニアライフを送りましょう、という激励の言葉でセッションを締めくくりました。



(所感) ユーモアが効いていて楽しめるセッションでした。また講師の村上さんは転職などを通して QA 室長にまでキャリアアップするという実績を持っているだけに、そのアドバイスは大変説得力がありました。

なお将来を見据えて勉強を頑張ろうというアドバイスは、参加者や実行委員のモチベーションを喚起し、WACATE の雰囲気をもっと活気づけることになりました。まさに朝一番にふさわしい熱いセッションでした。

(いせり)

実行委員も真剣に聞いていました。



◆ワークショップ3

上條 飛鳥 (WACATE 実行委員会)

「テスト分析はじめの一歩」

今回の WACATE2012 冬から実行委員となった上條さんのセッション。仕様書を汚しながらテスト分析や観点を考えるセッションです。



初セッションとは思えない堂々としたモデレート。

3色ボールペンを使って仕様書を汚すセッションです。



頭を使うので IBM の細川氏からお菓子の差し入れが！



これも珍しい、清水氏と細川氏の2ショット。



汚したあとは、各班ごとに意見交換です。
他の班ではどのようなことをやっていたのか？といったことを共有します。



実行委員長も語ります。



こうして二日目の朝は過ぎていき、ランチの時間となりました。

ランチはマホロバ名物・カレーバイキング！
残りの時間を駆け抜けるためにカレーで体力回復です！



○午後からは各班の発表

各班でどのような議論がされていたのかを発表していきます。普段はあまり人前じゃべらない若手エンジニアが発表する機会となっています。





さまざまな発表形式を見て、さまざまな意見に触れ、発表をしてみる—WACATEのワークはさまざまな要素があります。

◆クロージングセッション

松尾谷 徹氏(デバッグ工学研究所)
「CFD++」

今回のWACATEのクロージングセッションは、デバッグ工学研究所の代表である松尾谷さんによる「CFD++ ～デバッグ工学の夢～」と題したセッションが行われました。



聴いている人を飽きさせない軽快なトークと確かな技術的、確かなアドバイスに、聴衆が聴きほれます。

松尾谷さんはまず歴史を紐解きながらソフトウェア工学について解説。設計・分析手法などのエンジニアリングの整備を迫られるも、開発が複雑化していくうちに工学的アプローチだけでは力不足で実践やプロセス志向も重視されるようになった、というソフトウェア工学の揺れ動きについて説明しました。

次にそうした背景から、ソフトウェア工学には、従来の工学的アプローチを目指す要素と目指さない要素が共存していると説明しました。例えばプログラミングでは、とにかく早く試行錯誤して早く失敗し学習を進めるスタイルが有効とされ、緻密に分析・準備する科学的アプローチと少し違うとのこと。そして、松尾谷さんはそうした工学的アプローチを目指さない要素を指す例えとして、ブリコラージュ、シャーマン、ブリコルールという言葉を示唆。プログラミングはブリコラージュの世界でシャーマン的と解説しました。

その上で松尾谷さんは、シャーマン的アプローチは必要だが、価値判断は経験ベースのため、大規模・複雑な開発では混乱すると言及。最初のパートのまとめとして、シャーマンをエンジニアリングでサポートする重要性を説きました。

次のパートでは松尾谷さんが手がけた改善活動と、その1つであるCFDの解説を行いました。

まず松尾谷さんは過去に携わった開発で、品質会計の構築、原因結果グラフの導入、OMの開発を行なって成果をだしたが、原因結果グラフについては効果が頭打ちで、開発部隊に受け入れられなかったというエピソードを紹介。

そこでの考察の中で、「プログラミングは2値ではない制御の流れに注力する」「ロジックは仕様書に詳細に書かず、プログラミング時にプログラマが補完することが多い」といったものを見出し、2値のロジックを扱う原因結果グラフはプログラミングと合っていないと検討。結果、プログラミングで扱う「同値分割ベースの多値論理」「流れ図」の特徴を持つ手法として、CFDを考案した、という解説を行いました。そして次にCFDの概要と、単体テストでの有用性について紹介しました。

3番目のパートでは、設計との連結と題して、検証指向設計の解説を行いました。

松尾谷さんはまず開発で扱う論理は外部論理（ビジネスルール）と内部論理（機能の実現を支える論理）の2つに分けられると解説。そして論理を表現する手法として、決定表（ISO5806/JIS X0125）の解説を行いました。

次に決定表をベースにした論理の分析やブレイクダウンについて解説。外部論理を分析してコードに展開したり、無則を考慮しながらテストケースに展開したりする流れについて解説を行いました。

(所感) 広く深い経験に基づいた解説は大変示唆に富み、120分という長時間ながらかなり濃密で気の抜けないセッションとなりました。またエンジニアリングとブリコラージュ、設計とテスト、外部論理と内部論理、といった異なる領域全体を見定めて横断的なアプローチを取る考え方はとても刺激的で、テストの勉強だけでは得られない知見を学べる貴重な機会にもなりました。

(いせり)

◆クロージング

そして2日間の終わりを告げる、クロージングは副実行委員長の澤田さんから。



ベストポジションペーパーは、「テストについて気付いたこと」をわかりやすく書いた松木さんに決定しました。

◇そして最後は恒例の…



松木さんは今回のポジペが受賞すると思わなかったようで、とても感激している様子でした。

次回のWACATEでは、きっと松木さんの熱い熱いBPPセッションが聴けると思います！



WACATE2012冬、これがWACATE-Magazineがお送りする最後のレポートとなりました。

次からのWACATEの様子はWACATE-Blogなどでお届けする予定です！

また、近年では第三バイオリンさんによる@IT自分戦略のブログや

BPP受賞者によるgihyo.jpのレポートなどにWACATEの様子が掲載されていますので、是非そちらもご覧くださいませ！

WACATEはとても熱いイベントです。興味が出た方は是非、来て見て触れてみてください。

そこにはきっと素晴らしい気づきと、素晴らしい仲間がいます。

有難うございました！

(WACATE-Magazine 編集部)

See you Next WACATE!!

JaSST'12 Tokai Report!!

さる 2012/11/30、東海は刈谷・産業振興センターにて「JaSST'12 Tokai」が開催されました！



今年も豪華なプログラムにさまざまな地域から参加者が来ていました。

丸一日、テストを考えるシンポジウム！

それでは早速レポートさせていただきます！



◆◆◆オープニングセッション

◆◆司会

矢野 恵生氏

WACATEにも参加していただいている矢野さんの声で、JaSST'12 Tokaiの開始が告げられました。



◆◆実行委員長挨拶

奥村 健二氏

そして、実行委員長として登壇するのは我らがKENさんこと、実行委員長の奥村さん。



今回のテーマは「もう一度テストのことを考えよう。」
一日、皆さんと一緒に考えていきたいです。とキッパリ
言った言葉には、意志が感じられました。

◆◆◆基調講演

「ソフトウェアテスト、一番最初にやるべき大事なこと」
テストの目的をプロジェクトで共有する

湯本 剛氏



基調講演はWACATEのクロージングでもご登壇いただいた、
日本HPの湯本さん。
難しいテーマをアイスブレイクを交えながらわかりやすく
教えてくださいました。

◆◆ソフトウェアが直面する課題

ソフトウェアは以下の課題を抱えています。

- ・大規模化、複雑化によるテスト増大
- ・開発期間短縮かによるテスト工数不足

これには派生開発なんだから簡単にできるでしょ、という
場合もあるが、そのタイミングで出さないと負けてしま
う。ということもある。
適切なタイミングに出さないと売れない。

ちゃんとできていないと、市場で問題が上がる。
不十分なテストが発生する→運用後に問題発覚
→大幅な手戻り→本来の開発工数を確保できない
→不十分なテスト
…と負のスパイラルに入る。本来やらなければならない
ことができない。

テストを更に効率よく実施し、確実に品質を確保するに
は？→これが、直面する課題。

だからこういうイベントにも人が来るし、テストの勉強
会が起こる。

◆テストの開発がボトルネックになっている。

・テストは開発工数の約40%以上はテスト。
IPAのデータでもそのくらい。アメリカでも50%と言われている。「基本から学ぶソフトウェアテスト」でも45%と書いてある。

ボーリスパイザーの本ではミッション・クリティカルなものでは90%がテストになると書いてあるものもある。先日お客様を調べたところ80%やっているというデータも出てくる。ボトルネックになっている。

・ボトルネックであるテストの見直し
開発全体を良くする相乗効果がある。
ではなぜ、テストは増えるのだろうか？3つある。

◆欠陥の特性

NIST2002年のデータ。欠陥の多くは上流で発生する。
→いつ取り除くか？→最後の方で取り除く。
問題が出て治すコストは、早い段階から取り除くとコストがかからない。※バームとバジルの研究結果。

でもテストって最後の方で沢山やりますよね。
そもそもテストに工数がかかるやり方を私たちがしている。そもそもこういうものであるが、これを見直す必要がある。

◆開発の効率化

ソフトウェア開発を効率化しようとして努力している。
プロトタイプングやアジャイル、オブジェクト設計…ほとんどが効率化を考える手法。
昔はコードを書く時間が減った、という効率化したように見えた。

開発は部品を買ってくる、ユーティリティを買ってくる。すると開発工数が減る。で、テストが減っているか、という減らない。
ただ、全体的にはコストが減った、となる。

開発は、お金を稼ぐため、お客様に価値を提供するためにやっている。基本的なアプリはできていて、それを連携させて全体最適ができる、というのが今の時代。
今までは1つだけだったが、ハイバリューな製品としてソフトウェア同士を連携させたりすると開発は少ないものの、テストは増える。連携したときのテストも必要となるため、テストは増えていく。
→テストを手付かずにして新しいことにチャレンジすると、こういう問題が起こる。

◆テストのやり方

テストのやり方そのものが変わっていない。

○テスト開発技術

技術をもって作っているか。テストケースは開発するものであり、テストケースの品質を高めないといけない。

- ・的を得たテストケースを作る。
的を得ていないとテスト量が乗算で増加し、正しい判断を阻害する。
→狙いがわからない
- ・保守性の高いテストケースを作る。
一度作ったテストを再利用しやすくすることで、再利用時の工数を削減できる。

○テスト実行技術

テストのやり方。訓練が必要。

- ・人間の見落としは訓練しないと40%に及ぶ。
- ・見落としを最小限にする仕掛けが必要。
人間のミスを防ぐ必要がある。

テスト実行の自動化やテスト環境作成の自動化など JaSST'12 Tokyo の基調講演であったが、Microsoft は WinXP のテストはほぼ自動化している。
→新しい製品をテストする方に人にかかるために、昔のものはできる限り自動化している。

- ・自動化をすると実行工数が20倍～60倍の差になる。
- ・テスト実行工数の最大50%を自動化した事例もある。



○テストマネジメント技術

的を考えるのが人間。マネジメント不備が無理・無駄・ムラを引き起こす。

- ・テストの目的の明確化、優先度の継続的な見直し
(テスト戦略、テスト計画)
- ・資産の再利用に関する効率化
(構成管理やトレーサビリティ管理)
- ・バグ発見時のやりとり効率化
(正しい情報のやりとり)
- ・テスト結果の見える化に関する効率化
(モニタリング、報告書作成)

○テストのやり方とテスト現状

- ・開発技術ができていない
→人海戦術になる。考える時間を惜しんで人を投入。
- ・テスト実行技術ができていない
→手動によるテスト。データの準備、環境の準備、テスト実行、モニタリング、正解のジャッジ
- ・テストマネジメントができていない
→原始的なテストマネジメント。Excel を使ったテストケースドキュメント、ファイルサーバでのテストケース管理、メールベースのバグ報告、報告書作成が手動転記
→結果的に工数が増える。

こういうことを良くしていくことが求められていること。



◆◆ソフトウェアテストの全体像

テストは実行だけではない。実行だけで捉えていると、何を改善すべきかわからない。一般的な開発ライフサイクルの表現では、開発が終わるとテストができるようになっていく。

○開発ライフサイクル

計画→ビジネス要件→設計→開発→テスト→アプリ配備

「段取り八分」と言われるように、準備とマネジメントが必要。

まず、分析が必要。言葉にならないと識別ができない。原始民族などにもあるが、1と2とそれ以上しか知らないと、10の数ができない。識別して会話ができない。

「テスト準備」には計画、分析、設計、実装がある。これらを支えるマネジメントが必要。コスト、人、テストの資産管理、進捗管理、欠陥管理、基準評価報告、フィードバック…いろいろある。

これらは思いつきではなく、ISTQBで定義されている世界的にこういう定義がある。

◆テストを開発する作業

これで十分です、と言えるテストが良いテスト。良いテストをしっかりとるためには一連の流れをこなしていかなければならない。つまり、ちゃんとした設計をしましょう、ということ。

テストポリシーからテスト計画をもって、テストの目的を明らかにしてテスト分析をする。何故分析するのか？→対象を正しく理解するために分析する。正しく理解する行為である。できた「こういうことをテストしなければならぬ」というものを効率的に、正しく、上手くテストする。(設計する)その後、手順になる。

「テストの目的」というのは、テスト開発の最初のアウトプットである。だから重要。

◆◆テストは何故やるのか？

いろんな人にいろんな思いがあります。聞くと、いろんなこと言ってる。話し合うと、違うことを言い出す。

◆テストの必要性 (JSQTB より)

- ・Reduction—低減
システムやその文書をテストすることは運用時に問題が発生するリスク低減する。
- ・Evidence—証拠
当初の目的であるビジネスの成立、あるいは産業界の規則、もしくは開発時の契約を満たしているかをチェックする。
- ・Measurement—計測
テストを基にして、ソフトウェアの品質を測ることが可能。
- ・Effect—影響
テストプロセスを改善する中で、原因分析等を行いフィードバックする。

そのほか、テストの目的もある。

- ・欠陥を抽出する
- ・動作保証をする
- ・情報提供をする
- ・未然防止する

◆テストの必要性は目的と手段に分けることができる。

- ・目的—手段
品質を計測—欠陥計測、特性計測
証拠提示—合致確認
問題発生リスク低減—品質リスク管理、欠陥修正確認
システム品質を改善—欠陥原因分析、プロセス改善

- ・テストの目的：欠陥を抽出をすることで欠陥計測や品質リスク管理や欠陥修正確認をすることができる。
- ・その手段：表示の誤りが無いか、境界値とか0とかNULLなど間違えそうな入力データで出力が誤るか—このように階層構造がある。

ではなぜ、その目的が必要なのか？

◆想い

- ・品質計測に対する想い→「開発に携わる人だけでなく、誰が見てもわかるように品質情報を提供したい」
 - ・証拠提示に対する想い→「世の中のルールに沿ってビジネスしたい」
 - ・問題発生リスク軽減に対する想い→「利用する人たちにとって価値のあるシステムを提供したい」
 - ・品質改善に対する想い→「自分達の開発をより価値のある、やりがいのある仕事にしていきたい」
- 目的の先にある想いにも関係がある。

◆その想いは誰のため？

「世の中のルールに沿ってビジネスを成立させたい」
→認証機関の人、利用者（使う人）、利用者（メリットを得る人）

「開発に携わる人だけでなく、だれが見てもわかるように品質情報を提供したい」
→組織内の上長、監査、経営層

「自分たちの開発をより価値のある、やりがいのある仕事にしていきたい」
→自分たち、チームメンバー

—それぞれに相手がいる。

◆目的の階層構造・抽象手段の3要素

目的と手段は階層構造。

テストは品質管理の一手段。

品質管理はシステム開発の一手段。

どこを「立脚点」にして目的を階層化するか、が大事。

・混乱のもと

手段の話をするときに抽象的な話をしないと共有（シェア）ができない。でも抽象的なままだと仕事ができないので、具体化しないとイケない。

手段には必ず3要素が必要になる。

- ・分析：目的を理解する
- ・方式設計：抽象的なものを具体的にする
- ・詳細設計：実現する、次に繋げるための仕組み



◆◆テストの目的とは？

そもそもテストとは。SWEBOKのテストの定義。

「ソフトウェアテストとは、通常は無限に大きくと考えられる「プログラムの振る舞いの実行領域」から最適だと考えられる有限な「テストケースの集合」を選定し、所定の通りかどうかを実際に動かして検証すること」

テストは品質を写す鏡、

テストは鏡を使って知る「行為」

→自分で鏡を見て「自分イケてる」と思うこと。

よくないテスト（鏡）はひび割れていたり、指紋がべたべたついていたり。鏡単体では後ろは見れない。なので、後ろもちゃんと見ないとイケない。それは行為。一なの難しい。

・「何故、その行為をするのか？」

自分イケてるのは、何をしたいのか？

顔を見てもイケてないのはわかってる、ネクタイが曲がってないか、ズボン破けてないかを見たい。

・「どこを見たいのか？」

健康かどうかは鏡を見るよりも違うことを見ないとイケない。

・「見ることで何をしたいのか？」

別に目が2つある、ということを見てもわからない。

・「知ってどうするのか？」

どうしたいのか？

一いつも同じものを見るわけではない。

製造業はいつも同じものを作るために頑張るが、ソフトウェアはそうではない。まるで同じものは作れない。同じものはコピーで良い。

テストするのも製造業と違い「いつも同じになっていることをテストする」わけではない。

・そういう意味で、製造業のやり方は使えない。

・品質保証に対する誤解はここから来ている。

→いつも違うものを見ているのに、いつも同じ視点ではズレてるのではないか？

だけど学ぶところも多いので、そこを間違えてはいけません。

・製造業が効率をあげるためにしている工夫点はソフトウェア開発への重要な示唆。

→検証しやすく作る、IT化を推し進める。

→ただ「そんなこと考えて作ってられない」と言われることもある。

ソフトウェア開発のIT化ってどれだけ進んでいる？

→作る方が一番IT化されていない。

※ソフトウェア品質のホンネ。小井土さんの記事です。

テストすることも毎回同じではない。

何をみて、何を知りたいのか？知ったことをどうするのか？難しいのは、これが無いとどこまでやればいいのか？がわからない。

→だから、毎回考えなければならない。

◆最適とは？

「通常は無限に大きくと考えられる「プログラムの振る舞いの実行領域」から最適だと考えられる有限」

一何を持って「最適」とするのか？

・コードを実行してエラーにならなければよい。

・仕様に書いてある通りに動けばよい

・いつでもどんなときでも仕様に書いてある通りに動けばよい

・いつでもどんなときでも所定の速度で仕様に書いてある通りに動けばよい

・仕様に書いてある通りに動いたときに変な副作用（データ漏えいなど）しなければよい

・仕様に書いていないことも「利用シーンを想定し」うまく動いてくれればよい

・ソフトウェアを使うと当初の目論見を達成すればよい（商品売上〇%アップ）

→どれが最適？

一番最後だとは思いますが、漠然としすぎている。

最適を共有することが大事。「何を最適とするのか」を共有する。

知らず知らずのうちに考えている最適をあらためて整理し、共有する

○テストの目的の典型例

- ・欠陥の検出
仕様・要件に対する不具合を見つける
意図的な欠陥識別
- ・品質レベルが十分であることを確認
仕様・要件に対する合致性の確認
技術課題に関連して：解決策の妥当性確認・検証
リスクに関連して：リスクの評価、対応策の妥当性確認・検証
- ・判断のための情報を提示
信頼度の測定
- ・早期にレビューすることで欠陥の未然防止
要求に関連して：要求の妥当性確認、
要求の抽出（の促進）
要件・設計に関連して：要件・設計の妥当性確認・
検証

これらの話しを共有していますか？ということ。
これらがバラバラになっていると、後から後から増える。

◆何故共有する必要があるのか？

→ソフトウェア開発では通常、一人では行わないから。
規模の増大（人間の理解を超える）、コミュニケーション
ロス

◆現実の問題点

共有は難しい。

○開発の中で共有ができない

- ・開発のススメ方に対する考え方が違う。
外部設計、内部設計 vs 論理設計、構造設計
- ・そもそも共有しないまま開発を行なっている。
要求獲得、要求分析をせずに開発をする、
目的をはっきりさせずに言われたことしかしない。
ドキュメントを作らない；他人と共有しようとしなない、
身近な人にしか伝えない。
工数がかかりすぎるため、監視とコントロールを行わ
ない。共有したものがどうなったかは考えない。

○口頭だけで話しを決めて開発をしてしまう

- ・認識の齟齬が起こる
この○○にはAコンポーネントを使えばいい。
という話しをしたとして
Aさん：Aコンポーネントには□×の制約があるけど、
○○には関係ないはずだな
Bさん：ということは△△にもAコンポーネントが
使えるな

・あるお客様との会話

お客様：うちのチェックリストは細かすぎるから見てい
ない。テストする内容は口で伝えている。

ー市場でバグが出た時のどんなテストをしていたか
はどう報告するんですか？

お客様：チェックリストを提出します

ーチェックリスト、使ってないんですよ？

お客様：そうなのは本質じゃないですよ

→これはあまり良くないですよ。

◆共有するために大事なこと

- ・目的を具体化する
- ・目的の達成度合いをどう確認するかを決める
- ・全体像を提示する
- ・目的と手段の協奏

→テストの目的がその後どう使われていくかを理解しな
いとけない

- ・マネジメント課題
文書化：共有できない
テクニカルコミュニケーション：同じ言葉
交渉スキルの教育

◆具体化

テストの目的を2つの観点で具体的にする

- 見方
- 見る位置

テスト目的を満たす事が十分であるということは
全ソースコードを一回は動かしてエラーにならなかつ
た…など。

◆達成度合いの確認

十分かどうか？

◆全体像の提示

どこのテストをやっているのか？

他のところでテストしていないか？

WFでやっている場合はどのテストレベルか？

アジャイルの4象限のうちどこか？

4象限：TDD、機能テスト、シナリオ、非機能

◆目的と手段の協奏

目的と手段の両方が繋がっている

実現に無理がある：妄想

手段がばらばら：暴走

◆テスト設計

テスト目的を実現するための手段。

テスト目的とテスト対象を決めたらテストが実行できる
か？

・テスト対象に関する情報は全て仕様書に書かれている
か？

・仕様書に書かれたことを全てテストした方がよいの
か？

→時間が足りるか？それで十分なのか？

テストを設計すること=テストの目的を満たすテストケ
ースを作成すること

設計なしだと以下になりがち。

- ・気になるところ
- ・すぐ着手できるところ
- ・場当たりの

◆目的からテストケースへのつなぎ方

・JaSST'09 Tokyoの「テスト技法からテストメソッドロジ
へ」→智美塾

テスト開発プロセスの一例「ゆもつよメソッド」

計画、目的、テスト分析、テスト設計

◆まとめ

- ・ソフトウェアテストが直面する課題
→ソフトウェアテストは開発のボトルネック
- ・ソフトウェアテストの全体像
→テストに必要な活動を知る＝何をよくしていけばよいかを知る
- ・テストは何故やるのか？
目的と手段の階層構造
- ・テストの目的とは？
最適なテストケースの集合を選定する基準

◆Q&A

Q: テスト自動化 50%の事例があるということでしたが、業界はどこのものでしょうか？

A: 今回出した事例はエンタープライズ系です。

Q: テストとは何か？という話がありましたが、
テスト＝動かして検証するだけの行為、ソフトウェアテスト＝全体として実行だけではなくて設計 実行 報告まで、という言葉として使われていたように思うのですが、双方のつながりはどうなのでしょう？

A: テストとテストリングは英語の文献では厳密に分かれていて

テストリング＝実施する行為。

テスト＝テスト自体。テストケースの名称です。

今回の「全体の話」は厳密に言うところ「テストリング」です。テストリングの方が意味が広いです。行為。アクティビティです。

Q: 最初の方にテストの工数＝40%占めるというのありましたが、今回湯本さんが説明されたことを全てやると考えるとどのくらい減ると考えていますか？

A: HPの事例ですと、テスト工数が2割減ると言われています。30%くらいかも知れません。XDDPなどで実施すると、もう少し減らせるのかな、という感触です。

◆感想

説明が難しいテーマを、とてもわかりやすく解説されていたかと思えます。キモとなったのは、目的と手段の話。ここは少しむずかしいという意見もtwitterなどで見かけましたが、目的と手段を意識することはとても大切であり「どこに立脚しているか」ということを共有するのはとても大切なことです。

これらを上手く説明できる方はとても少なく、そういった意味で今回このJaSST'12 Tokaiに参加できたことはとても自分のためになりました。

素晴らしい講演でした。

◆◆◆ポスターセッション

◆◆Neusoft(スポンサー様)



◆◆SHIFT(スポンサー様)



◆◆Fujitsu(スポンサー様)



◆◆田中学二さん(独自のテスト管理 Tool の紹介)



◆◆NPO 法人 ASTER(活動の紹介)



◆◆TEF 東海合宿(活動の紹介)



◆◆バグ票ワーストプラクティス (活動の紹介)



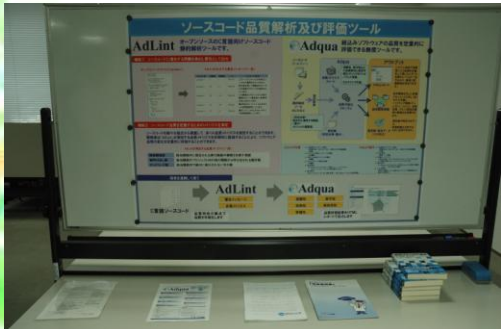
◆◆TEF 東海原因分析道場 (活動の紹介)



◆◆小池さん (書籍の紹介)



◆◆AdLink (ツールの紹介)

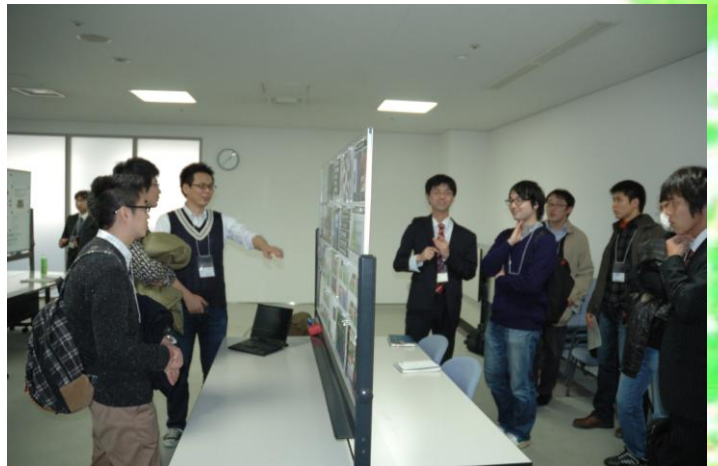


◆◆WACATE (活動の紹介)

そしてわれらがスター☆による WACATE の紹介です。



沢山の参加者の方がポスターを見ながら、質問等をお交わしていました。



◆◆◆経験発表

「アジャイルなテストの見積りと計画作り」

きょん(SCMBootCamp)

WACATEにも参加してくださっているきょんさんの発表です。



Webのテスト(Server Side)を担当している。

開発環境改善。Groovy, SCM, Test, Agile, CD, 関数/証明プログラミングなど。継続的インテグレーションなど。

本発表は私の独断と偏見と体験(コミュニティは無関係)。

◆What's Agile?

アジャイルはプロセスではなく、スタイルです。

アジャイルの基礎は、アジャイル宣言とアジャイルの12の原則。

Haskell が関数プログラミングスタイルの1つの実装であるように、XP、Scrum はアジャイルの1つの実装。

◆Background

テストエンジニア1年生 GUI のない Web アプリでサーバサイド

◆Problem

- ・テスト観点ってなに？
 - ・テストの見積りが難しい。
 - ・品質ってなに？
 - ・テストはどうやって区切るの？
 - ・どこまでテストすればいいの？
- 何かできないかなあ、と考えた。

ProductOwner, Developer, Test という役割分担。

作りたいもの+品質モデル+設計

全体戦略ができれば、あとはスプリントで作る。スコープを決めてインクリメンタル。

エンドツーエンドで作る。戦略へのフィードバックをずる。

アジャイルで大事なこと

「方針やフィードバックをどれくらい早く共有するか」
個人的意見。アジャイル=早いフィードバック

◆チームコラボ

- ・Analyze Requirement
- ・フィーチャー
- ・技術的アーキ
- ・スケジュール
- ・チームのリソース
- ・クライアントのリソース

◆Use Tool

- ・5W2H
 - ・マインドマップ
 - ・フィーチャーボード：機能一覧だったり要求とリスクを分けたりする。
 - ・テスト観点モデル
- これらを使って共有していく。

◆Share

- ・テストレベル
 - ・品質モデル
 - ・テストタイプ
 - ・技術的リスク
 - ・市場リスク
- これらを共有している。

◆テストレベル

コミットステージ：ユニットテスト

ストーリー受け入れ：ハッピーパス（一機能が一番うまく行ったときのパス、王道。）

→だいたい1ストーリーに2-3ケース

結合：テストエンジニアによるテスト

個人的にこれがやりやすい。

自動化の範囲はプロジェクト毎に違う。

基本的には弊社は自動化以外のテストは認めないスタイル。

◆品質モデル Quality Model

- ・ISO9126
- ・FCM(Walters & McCall)

○ISO9126

- ・わかりやすそう。
 - ・型から入る。TypeではなくForm。扱っている例が多いのも良い。
- これをベースに考えた。が、どんな品質が必要か考えてみたが、自分にはあまりわからない。製品に結びつけるのが難しい。そこで、経験。NET, Android は経験が無い。

○ISO9126+Team

仕様が曖昧だけと作らなければならない。

開発者が思う次のような不安点を分類した。

- ・仕様が曖昧な部分のテストの漏れ
- ・テストが困難故に単体でしかテストしていない

ProductOwner が思う次のような不安点を分類しなおした

- ・運用時
- ・確認できていないもの

ISO9126+Team。この取り組みは良かった。

◆Risk

- ・設計
- ・ビューティフル・コード
- ・パフォーマンス
- ・バグ修正→修正にどれだけ時間がかかるか
- ・顧客のビジネス影響
- ・連携サービス

◆Effective by share

- ・テストの優先順位の意識付け
 - ・どの品質を対象にしているのかの意識付け
- まずはマインドマップ+ISO9126で議論をスタートするチームが増えた。

◆◆見積り

テスト観点＝何かを実証するアプローチのこと、と定義します。テスト観点ごとにテストするとやりやすいかもしれないと直感があった。しかし観点を列挙すると150個以上になった。相対見積りをした。やってみてすぐ悩む。何を見積もればいいんだろう？

- ・規模？→Agileと同じようにする？なんだろう？
- ・複雑さ？
- ・時間？
- ・他のもの？

○できそうなもの

- ・パラメータ数
- ・難易度
- ・どれくらい組み合わせるか

○規模

- ・テストポイント＝それぞれを掛け算していった。
 - ・適当に割り振ったものをポイントとして掛け算した。
- 6000テストポイント＝何日かかるのかよくわからない。

◆Flow

- ・テストアーキテクチャ構築＝あくまでモデルであり、スナップショットなので不完全。
- ・テストポイント見積り
- ・テスト戦略策定
- ・1日だけうすくひろくテストを実施する。
- ・テストポイントの歳見積り
- ・戦略再策定
- ・テスト実施

実際に看板みたいなものを作った。TestBoard。
技術的難易度とビジネス優先度を軸にマッピング。
パラメータが多いとアトミック性が高い。＝分割しやすい。
分担数が増えるとバグの混入率が上がる。テストの独立性が上がる。



◆うすくひろくテストする

調査的テスト、探針？ちょっとわからない。できるだけまんべんなくテストする。できるだけ、まんべんなくテスト観点を見る。Test Boardを1日で1周することでだいたいどれくらいかかるか、がわかってきた。

- ・1st=1500tp/1week
- ・2nd=2000tp/1week
- ・3rd=1800tp/1week

○見積り可能なテスト

- ・見積り可能なテストはテストングが導いてくれる

○依存関係

スプリントのテスト戦略と全体のテスト戦略、フィーチャ、品質モデル、リスクより早く、より軽く、より観測しやすくする。変更があることを前提に頑張る。Test Arch→Sprint→Implemented Test Architecture
どンドンテスト作る→バグが見つかる→フィードバックする。
一継続してやっていきたいと思う。それが良いと思う。

- ・テスト観点＝共有
- ・見積り＝相対見積りでざっくりとできない範囲
- ・品質ってなに？＝ISO9126+Team
- ・テスト区切り＝観点フィーチャ単位などを実施した。制約ややりやすさで変わる。
- ・どこまでテスト？＝テストしてからでは遅い。リリースのDoneを最初に共有し徐々に修正していく。

○続けたいこと

- ・品質モデルの共有
- ・チームのレビュー
- ・相対見積りの活用

○課題

- ・ボードの完成度

○できそうなこと

- ・テスト観点のネスト構造についての実践（テストバスケット？）
- ・テスト観点のリファクタリングや設計パターンの構築

○気づいたこと

- ・アーキテクチャは自分が思った以上に作りこむ必要がない
- ・テスト観点モデルは徐々に成長させることになる
→独立性と合成性が高いテスト観点を考える事が重要。

◆質問

Q. Architectureのイメージ
A. モデル図など。

Q. Implemented Architectureのイメージ
A. コードから自動生成する

Q. テスト観点の共有はどうやってるの？
A. Boardを物理的に用意している。Architectureを見てもらう。

◆感想

LT形式の発表はテンポが早くついていくのが大変でしたが内容は洗練されていてとてもわかりやすく、考え方は割りと王道でした。

特に「一日だけうすくひろくテストする」のくだり。恐らく現場ではそういうことをしているところは多いかな、と思いました。

タイトルの見積りの話よりも「どうテストで効率よく早くフィードバックをかけるか」という点の話がかなり面白かったのですが、見積りについて独自の「テストポイント」という概念をもって実世界とのすり合わせをしているあたりが話のキモでした。

今後、ポイントの算出基準と数回のうちの統計的基準のようなものが出せるようになると、ますます精度が上がリ、素晴らしいものになるように思いました。

◆◆◆テスト設計コンテスト東海地域予選会

JaSST'13 Tokyoにて決勝が行われる「テスト設計コンテスト」今まで各地域にて予選が行われてきましたが、その東海地区戦です！

SESSAMIが出している要求仕様書をもとにテスト設計をしていただき、JaSST'13 Tokyoの本戦出場テスト設計のNo.1を決めるコンテストです。

◆審査委員

審査委員長

鈴木 三紀夫

審査委員

秋山 浩一

西 康晴

湯本 剛

吉澤 智美

奥村 健二

東海地区では2チームが予選出場。各チームの発表をしていきます！

◆◆士ーSABURAIー



全体コンセプト

今までは仕様書から設計していた。

- ・第三者が理解可能な分析・設計
- ・無機質なテスト設計だけで終わらせない

分析と設計の流れ

- ・分析

DFDによる全体仕様の把握

品質特性によるテスト分析

ユーザの性格を考慮した要求分析

ユーザの環境を考慮した要求分析

- ・設計

分析結果から機能に合った設計手法にてテスト設計を行う。テスト設計では、できるだけ表や図を使う手法でテスト設計を行う

○分析の詳細

DFD。こちらの段階からトレーサビリティを意識。

→検証漏れをなくす。

品質特性によるテスト分析。独自の分析？FV表？

ユーザ性格とユーザ環境考慮による分析

→なぜこだわったか、というのは

AさんBさんという新人にテストをお願いした。

AさんとBさんと全く同じ時間、同じ製品を操作したにも関わらずBさんは全く不具合を発見できなかった。

→性格と生活している環境によって人によって差異が生まれる＝価値観の違いか。

性格と操作の洗い出し

指示に従う＝すなお

指示を無視する＝いじっぱり…など。

他にもせっかち、おだやか、きまぐれなど。

性格を観点に入れて検証内容を考え、その内容を拡大する。品質特性で考えたものと性格と環境を考えたものをマージする。

○設計

機能テスト

動作条件テスト

組み合わせテスト

- ・状態遷移図

C1パスカバレッジ100%にて、全てのものを通るようにしている。

→状態遷移表を作成する。一巡するだけでは不具合を発見することはできないので、1スイッチカバレッジ表を作成する。※2回目の遷移までを確認する

- ・CFD

→デジジョンテーブルの作成

- ・AllPair法

シナリオテスト実施の目的

機能テストだけでは出しきれないもの。

機能上の状態と外的要因の状態の2つで考える。

禁則マトリクス→ありえない状態を排除する目的。

作成した状態に対して操作を実施していく。

→各種性格を組み合わせで実施していく。

→せっかち＝瞬断や繰り返し

実際のテストケース

操作と判定基準

○実施効果

- ・仕様書コピーの設計書より、レビュー時間を低減することができた。
- ・レビュー時の指摘件数が減少し、テスト設計書の信頼感が高まった。
- ・ユーザの性格や環境を考慮できたため、実施者の依存度を下げることができた。

○課題

環境と性格を組み合わせる新たなケースを作成したかったが、テストケースが増大してしまうため断念した。今後、ユーザ観点についてのものも作っていきたい。

◆◆MKE98



目的：今までの自分たちのものを改善していきたい。
要求や仕様に関する考慮漏れに気づけることを確認する。
私がやってきたもの＝仕様からテストケースからそのまま抜き出す。
→仕様に書いてないことは？→経験。あとひらめき。→これだけでは新人や違う製品のテストはできない。
→何か方法で補えないか？

・前提シナリオ
ソフトウェア設計 ソフトウェアテストまでを請負うプロジェクトである。
まだ開発に着手していない。

・テスト設計プロセス

要求仕様書

機能分割→機能一因子一覧

同値分割→FL表

状態遷移設計→状態遷移図、状態遷移表

テスト観点抽出→テスト観点一覧

テスト設計→テスト観点の一覧

品質特性、テストタイプ、テストカテゴリ、

検証内容、テスト技法、テストレベル

機能と因子をまとめて一覧を使う。

関係性が高いので組み合わせをするときに参考にできる。

観点一覧

品質特性一テストタイプ一テストカテゴリ一検証内容一

技法一テストレベル

テストカテゴリに対して検証内容、テスト技法、テストレベルを定める。

○アピールポイント

仕様書に書ききれない情報を抽出・整理し、よくある考慮漏れを減らす。

減らしたい考慮漏れ

- ・品質要求：速度、メモリ使用量、使い勝手→非機能的な要求
- ・利用目的：どこで使うの？
- ・異常：想定外の異常な状態や入力値など。

もっとよく考える。考えろと言われてもどういう視点で考えるのか、が重要。

考慮漏れを狙って減らす

- ・品質要求：テストカテゴリ抽出
- ・利用目的：6W2H
- ・異常時：HAZOP

6W2H, ガイドワードなど、軸を決めて分析を行う
→体系的に考慮漏れの検出ができる。
分析結果が残る。レビューや再分析もできる。

◆工夫点1

- ・品質特性を軸にテストカテゴリを抽出・整理
- ・過去の経験も入力にする
→流用できる考え方は流用する。
- ・テスト設計の成果をもとに、テストカテゴリは随時見直し
→テスト設計に気づくこともある。

◆結果1

多数のテストカテゴリを抽出できた。
テストカテゴリのうち、8割は要求仕様書以外から抽出した。
・車載制御ソフトのテストカテゴリはほぼ全てを活用できた。
テストケースとなると流用がしづらいが、カテゴリだと流用しやすい。

◆工夫点2

- ・6W2H
マインドマップで発散させた。

◆結果2

仕様には明確に定義されていない利用目的が抽出できた。

◆工夫点3

HAZOP ガイドワード
No, More, Less, As Well as, などからのズレ

- ・手順
ヒータの機能、因子一覧からガイドワードをもってイメージする。
→沸騰状態でも100℃にならない、というようなもの。

◆結果3

記載がない想定外の状態や入力値を抽出できた。
異常状態を34件抽出できた。

ポットの傾き、揺れなど。
想定外を見つけ、想定内にするためにガイドワードは有効と感じた。

◆評価

仕様書に明確に定義されていないことの検出数
仕様 QA の発生数
→いろいろ見つかる楽しいことがわかった。

◆まとめ

設計をすることで考慮漏れに気づけることがわかった

◆残課題

完了の判断が難しい。
手法を使うためにはトレーニング必要と思った。
組み合わせに関する考慮漏れを減らす方法は未検討。

一両チームの発表の終了後、審査員からコメントがありました。

◆◆各審査員からのコメント

◆奥村さん

MKE: 人の差を埋めるためにガイドワードを使っているのが印象的。
分析→詳細になっているので、方式が見づらかった。成果物はよくできていたのが印象的

◆秋山さん



レベルが高く審査が難しかった。

・ SABURAI

ポケモンは性格が25種類あるので、もう少しあるかもしれませんw
いじっぱりには操作性が関連するのではないかな。機能に対して目的になっている。

・ MKE

考慮漏れがポイントだったと思うがテストで考慮すべき考慮漏れとテストで考慮すべき考慮漏れがあるのかな、と思いました。
必要なところは開発と協業するとよいか、と思いました。

◆鈴木三紀夫さん



ユーザ特性を打ち立てているのはあるが、性格に落とし込んでいるのは新しい軸だな、と思う。

昨年の TETTAN の成果物を非常に研究しているのではないかと思った。DFD で全体を把握したり、CFD を使ったり。→これを取り込もうというのはとてもいいな、と思いました。

テスト全体を見ようという心がけがひしひしと伝わってきた。テスト対象を吟味したうえで全体を見ようという意志が伝わった。トレーサビリティを重視している。

◆湯本さん



両方ともレベル高いです。

SABURAI さんはテスト対象に対する指摘が足りないかなと感じた。厚みは全体的にわかるのですが厚みがわかりづらかった。

MKE98 さんは、テストカテゴリを使ってというのは車載の開発を適用させると、他のドメインでも使えるというのはよかった。開発やっている人たちも使えるのでは、と。テストケースを作るのは、それから更に一つクッションが必要と思いました。

◆吉澤さん



・ SABURAI

言葉の間違いなどが多いのと、誤字脱字。用語等については配慮いただきたい。

・ MKE98

テスト設計にどう繋がっているのか、がよくわからなかった。HAZOP がどう繋がっているのか。
双方仕様書から落とすところからあまり逸していないかな、と。ただ、双方ドキュメントのレベルが高い。

◆にしさん



エンジン：アーキテクチャ

足回り：設計

と例えると、

・ SABURAI

エンジン：ノーマル

足回り：良い

・ MKE98

エンジン：高い

足回り：ノーマル

それぞれが良いと思います。予選と同じクルマでは勝てない。審査は面白かったです。非常に勉強になりました。

◆◆感想

どちらもレベルが高く、ドキュメントは SABURAI が素晴らしい、発表と内容については MKE98 が素晴らしいという白熱した対戦となりました。

このように前年度チャンピオンの成果物を意識して洗練していくあたり、ET ロボコンのようになってきて、凄いことだな、と思いました。

数年後にはこのコンテストのレベルは格段に高くなるのではないのでしょうか。

◆◆◆SIG

JaSST Tokai 名物の SIG。議論をする時間です。

JaSST の中でも SIG として複数テーマについて参加者が語り合うことができるのはこの JaSST Tokai だけです！

SIG とは、1 グループ 10 人程度にわかれて各テーマに応じた議論を行う「Special Interest Group」です。

各テーマごとに各自で抱えている悩みや課題についてざっくりばらんに情報・意見交換を行う場です。参加者同士で情報・意見交換することにより問題解決の糸口を見つけたり、新たな気づきを得ることを目的としています。

他の会社の方と交流をしながら議論ができる貴重な時間。各テーマ全てを紹介することは難しいのですが、簡単などのようなことがお話されていたのかをご紹介します！

◆◆テストエンジニアだからこそ気づけるテスト仕様書と報告書の書き方(オーナー：山本さん)

知識がある人は論文は書けるが、いい開発はできない！そのため大学では on the job learning とよく言うけれど…いい開発いいテストをするには何が必要か？

いいテスト仕様書やいいテスト報告書を書き続けるといいテストができるのではないかな？

他者の経験を分かち合ってシミュレーションをしてみよう！

ということで、仕様書や報告書の書き方について、「困っていること」「工夫したこと」をそれぞれワークショップで発表し、他者からアドバイスを貰ったり、与えてみたりと経験共有をしてきました。

実に面白いヒントがいっぱいでしたよ！一見違う役割やドメインであっても悩みは共通だったり、逆に解決のヒントを貰ったりして、積極的な議論が展開されていました。

◆◆探索的テストから考える現場の工夫(オーナー：都築さん)

まずは「探索的テストとは？」という解説をしていたのだあと、実際の現場で良かった！と思ったことをまとめてみる議論がなされていました。

中には「バグのパターン化」や「5S」の話。

どこを探索するのか、といったことが議論されていました。

◆◆女性エンジニアのキャリア悩み相談室(オーナー：越智さん)

テーマに紐付かれるように、テーブルの雰囲気があっけらか♪

しかしながら、女性同士悩みを語るだけでなく、そういう悩みに対応する必要がある男性の方も参加されていました！

もっとも議論がなされたのが、やはり出産・育児関連による仕事との両立についてでした。

その他、やりがいやキャリアに関することについてもお互い情報共有したり先輩の経験談を伺ったりしました。

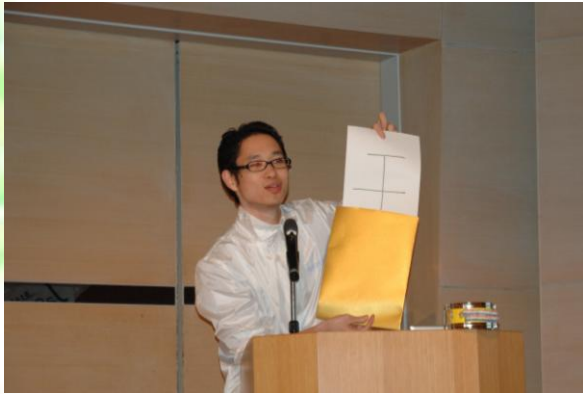
◆◆◆クロージングセッション

濃い一日もとうとう終了となりました。

まずは、テスト設計コンテストの結果発表が、司会の矢野さんからありました！

◆◆テスト設計コンテスト結果発表





全国出場チーム：士ーSABURAIー

東海地区は非常にレベルが高かったです。

東海予選会ほど、審査委員の中で意見が割れたことは無いです。非常に僅差です。ある審査員は一点差。Webに公開している基準に合わせて採点していますが他の予選会に比べても、レベルが高かった。ドキュメントの質が高かった。そのままお客様のところに納品できるのではないかというくらいの質。それだけ、作ることに對してきちんとしているカルチャーかな、と。また2チームともに利用者観点で考えられていました。他のところも同じようにユーザのことを考えてお仕事されているからこそと思います。

ドキュメントの質が高いがゆえにテスト設計の意図がわからない。ということもあります。開発以上に、テストのドキュメントは未成熟。いろいろな会社の人が試行錯誤している。JaSSTのWebサイトを見てわかるように過去の設計ドキュメントを見ても、非常にばらつきが多い。余計に意図がわからない。

北海道：Yuki Da RMA
東京：てるぼう、Shelly、三浦マグロ部、チーム nemō
関西：たからづかてすと団
書類：ZUNDA
東海：士ーSABURAIー

SABURAI チーム、本戦出場権が得られましたが各チームが偵察にきています。このままではおそらく勝てません。是非、これからも精進していただきたい。



◆SABURAI チームコメント

貼ってあるドキュメントを見てメモをしていたので盗まれるかも知れませんが、これからもさらにいろいろなことを取り込んで、優勝を狙いたいと思います。

そして最後に、本日一日を振り返っての実行委員長挨拶です。

◆実行委員長挨拶

東海代表として出場するチーム士。
是非、本戦も優勝していただきたい。

一日やってきました。我々としては「テストの目的ってなんだろう」とこれからも皆様に考えていただきたいと思います。

JaSST Tokai はたくさんの粒度で発表できる場を用意しています。



是非、皆さんも小さなことでも良いので発表いただければと思います。
外からのフィードバック、たくさんいただけます。SIGオーナーも募集しています。

本日はみなさま有難うございました。

◆◆全体の感想

湯本さんの「テストの目的」についての素晴らしい講演に始まり

きょんさんの「テストの見積り」があり、テスト設計コンテスト予選に加えて毎年恒例となった参加者自身がアウトプットするSIG、と今回もJaSST Tokaiは濃厚なテストの日でした。

たくさんの気づきと交流、情報の波に触れた気分です。また、後日行ったひつまぶしも美味しかったので大満足！のJaSST Tokai、次は是非みなさんもお参加くださいませ！

(WACATE-Magazine 編集部)

Software Test Topics

当番：コヤマン (Magazine 編集部)

こんにちわーw コヤマンどえす。

【イベント】JaSST' 13 Niigata

<http://www.jasst.jp/symposium/jasst13niigata.html>

開催日：2013年3月15日(金)

開催場所：万代島ビル 11F NICO 会議室

参加費：

1日券：2,100円

情報交換会：無料

参加申し込み締め切り：2013年3月12日(火) 18時

基調講演はガイオ・テクノロジー(株)のエバンジェリスト、大西 建児氏です。

講演概要を見る限り、仕様ベースのテストをするうえで十分でないテストベースに対してのテストについての上手な進め方についてお話いただけるようです。

完璧なドキュメントは無いですし、十分な品質のドキュメントがあることは稀ですね。

なかなか仕様が決まらずにすするとスケジュールが後ろに…なんてこともよくありますよね。

そんな中でどうしていったらよいのか。大西氏の豊富な経験からさまざまなヒントを聞くことができそうです。これはとても気に入りますね！

また、チュートリアルとしては「[マインドマップから始めるソフトウェアテスト](#)」の共著で知られる鈴木三紀夫氏による「マインドマップを用いたテスト要求分析・設計エクササイズ」！

あのマインドマップ本の著者自ら、マインドマップを用いたエクササイズとはとても豪華ですね！

たくさんのお気づきを得られそうです！

【書籍】Android アプリ テスト技法

<http://amazon.co.jp/o/ASIN/4798037044/wacate-22/ref=nosim>

単行本：217ページ

出版社：秀和システム

ISBN-10：4798037044

ISBN-13：978-4798037042

Android テスト部による、Android アプリのテスト手法を解説した本が出ました！

なんと、次回 BPP 発表を行う松木晋祐氏や WACATE 実行委員である井芹洋輝氏も執筆陣に名を連ねています！

具体的な手法の解説だけでなくサンプルファイルのダウンロードまでできるそうですよ！

今後モバイル化が進んでいく中で、是非手元に欲しい一冊ですね。

【イベント】ソフトウェア品質)シンポジウム 2013 (SQiP2013) 一般発表募集

<http://www.juse.or.jp/software/217/>

募集締切：4月22日(木)

シンポジウム開催：9月11日(水)～13日(金)

会場：東洋大学 白山キャンパス(予定)

ソフトウェア品質に関する3日間連続、日本最大級のシンポジウム、SQiPシンポジウムの一般発表の募集がされています！以下のテーマが募集されています。

- ◆要求やシステム開発などに関連した観点
要求工学、要件管理、システム開発と業務改善、ベンダとの協調、保守・運用、SLA、SLM、ITIL、など
- ◆ソフトウェア開発技術の観点
アジャイル開発、派生開発・プロダクトライン、形式手法、モデル検査、PSP/TSP(Personal Software Process / Team Software Process)、設計品質、アーキテクチャ品質、コーディング規約・作法、など
- ◆マネジメント技術の観点
オフショア開発、プロジェクトマネジメント、リスク管理、モチベーション・コミュニケーション、パートナーとの進め方、スキル・キャリア、人材育成、教育/トレーニング、小集団活動、など
- ◆品質管理・テスト技術の観点
テスト技術、レビュー、デバッグ、プロセス改善、CMMI/ISO15504、監査、QMS 構築、構成管理、メトリクス、不具合管理、受け入れ検査、品質の定量的把握(予実管理・EVM)、など

投稿は以下のカテゴリで行われます！

○経験論文(論文)

ソフトウェア品質に関する新たな提案、既存の方法・技術の実践などを報告する論文です。採録された論文に対しては、専門家によるレビューが行われ、論文をよりよくするためのコメントを差し上げます。

○経験発表(パワーポイントスライド)

実践活用事例、ノウハウなどをまとめた発表です。また、現在取り組んでいる活動の速報、問題提起等もお待ちしております。原稿の形式は、スライド形式のみです。

また、発表を迷っている方には…以下の特別企画も用意されています！迷われた方は是非参加してみてくださいはいかがでしょうか？

◆SQiP2013 委員会特別企画

「投稿応援フォーラム：社外発表のススメ」

<http://www.juse.or.jp/software/443/>

このコーナーでは、書籍やイベントの情報を募集しています。

例えば、「来月勉強会とか OFF 会とか開催するんだけど掲載してくれない？」といったものでも OK です。

開催地も、日本全国津々浦々、どこでもかまいません。

情報をお待ちしております m(_ _)m

開運

源太郎のソフトウェアテスト占い

※「ソフトウェアテスト占い」とは…
日本の伝統的な暦で知られる気学(九気性)をもとに、日本古来の統計学とソフトウェアテストで培った時代の統計学を組み合わせ考察された占いである！(笑)

	生まれ年	1月の運勢(1月5日~2月3日)
一白水星	S38/S47/S56/H2	本命星の影響と震宮に会しているため、 運気は好調! 震宮の影響を大きく受けるため物事をスタートするには吉なり。また 閃きが冴える月 になるであろう。ただし、 暗剣殺のため軽率な行動は逆運を招く と心得よ。 口は災い のものと成り。
二黒土星	S37/S46/S55/H1	本命星の影響と巽宮に会しているため、 運気停滞気味 なり。年盤で 五黄殺 を受けているので、万事 慎重かつ冷静な行動が吉なり! ただし 縁を紡ぐには良い月 のために、 ひらめき を信じて進むべし!!
三碧木星	S36/S45/S54/S63	中宮に会しているため、 周囲から注目される ことも多くなるであろう。また年盤の本命星と相克のために、 運気は低迷気味 なので 無理は禁物 なり。 独断的になりやすい 時期なので、 慎重・冷静 を旨に 俯瞰的に行動すべし 。
四緑木星	S35/S44/S53/S62	乾宮に会し本年が六白金星のため、 乾宮の影響を大きく受ける であろう。乾宮とは相克かつ年盤で暗剣殺を受けているので、 運気は低迷気味 なり。「風」の弱点が強く出るために、 足元が不安定になる ので、 慎重に行動することが吉なり 。
五黄土星	S34/S43/S52/S61	本月は本命星と相克なれど、兌宮に会し年盤中宮と相性のために、 運気は低迷気味 なり。兌宮に会しているために、 小さな喜びごとが重なる であろう。ただし 浪費をせず に 日々感謝を忘れず に進むことが、吉と心得よ。
六白金星	S33/S42/S51/S60	本月は本命星と相克なれど、艮宮に会しているために、 運気はほどほど なり。年盤で九紫と同会しているために、 精神的なものに大きく影響を与える であろう。 ストレスがたまりやすい 時期なので、 心身の健康を心掛ける ことが肝要なり。
七赤金星	S32/S41/S50/S59	離宮に会して月盤中宮とも相克なために、 運気は低迷気味 なり。 問題が表面化しやすく、争いごとに発展しやすい 時期なので、 発言には注意すべし 。熱しやすく冷めやすく、 精神定期に不安定 なので、 慎重な行動が吉なり!
八白土星	S31/S40/S49/S58	月盤中宮と相克なれど、坎宮に会しているために、 運気は低迷気味 なり。業務で ハイリスク・ハイリターンを狙う と、外部から水を差されて、 裏目に出る であろう。ただし 勉強・スキルアップするには、良い 時期なり。
九紫火星	S30/S39/S48/S57	年盤中宮とは相克なれど、月盤中宮・年盤で同会している三碧・坤宮とすべて相性なために、 運気は上昇傾向 なり。坤宮の影響により、 物事の動きが遅くなる が、来月以降の 準備期間と心得、努力を忘れることなかれ 。

月盤 三碧木星				年盤 六白金星				
南				南				
東 暗剣殺	2	7	9	東	5	1	3	
	1	3	5		西 五黄殺	4	6	8
	6	8	4			9	2	7
北				北				
				西 暗剣殺				

◆今月のチェックポイント!

今月は「三碧木星」が中宮となり、【雷】の卦象の影響により「拙速」「正直」「口」がでる月です。今年「六白金星」中宮のため、「独断的」「頑固」という面を強く出ますので「論争」「憤慨」とならないように、には注意すべし!! 「独断的」「頑固」から思慮不足となり「拙速」「正直」「口」がもとで「論争」「憤慨」にならないように「慎重」かつ「丁寧」を心掛けることが、運気を高めるでしょう。

★今月のラッキーテストアイテム

三碧は「時代の先端」の特徴がある星です。今月は「時代の先端」に行くことが今後のスキルアップを促進されるでしょう。

ソフトウェアテストで「時代の先端」と言えば・・・ そうだ! JaSST Tokyo へ行こう!!

お知らせ

WACATE-Magazine は次が最終号となります！

今回は「JaSST' 13 Tokyo レポート！」を予定しています。

来たれ！投稿戦士！

WACATE-Magazine では“ソフトウェアテストや品質に関する記事”を常に募集中です。
来たれ！投・稿・戦・士！！

WACATE-Magazine はボランティアベースで発行されています。したがって原稿料は出ません。※ノーギャラです。
投稿いただいた原稿の掲載可否、掲載時期については編集部で決定させていただきます。
特定の商用ツールやサポートなど営利色が強い原稿は原則として掲載いたしません。
レイアウトなどは全て編集部で行います。また、特に戻り確認などは行いません。
単なる論文は掲載しません。紙面の雰囲気を読んだ、イイ感じな文体や内容をお願いします。
公序良俗に反すると思われるものについても掲載しません。

以上を了解した上で、覚悟完了！な方は WACATE-Magazine 編集部「magazine@wacate.jp」まで、是非ご連絡ください。
折り返し、担当より要綱をご連絡いたします。
楽しい、そして役に立つ紙面にするため、是非ご協力いただければ幸いです♪

☆その他、WACATE-Magazine では以下の情報をお待ちしています☆

- 書籍情報(オススメ書籍情報も可)
 - イベント情報(ソフトウェア/品質/テスト関連)
 - 勉強会情報(ソフトウェア/品質/テスト関連ならなんでも)
 - オフ会情報(ソフトウェア/品質/テスト関連)
 - デザート情報(甘いものならなんでも。イカス！やつで。)
- などなど、お待ちしております♪



来たれ
次世代の
戦士達よ！！

WACATE実行委員募集中♪

★★★ WACATE 実行委員募集中 ★★★

編集後記

いのみん★花々咲き乱れる春到来！そして花の粉も！かひゅーん(花粉)かひゅーん(TwT)
ヨネビ★スノボで転びまくって、アザだらけ>>
コヤマン★かふんがあらわれた！めのかゆみがとまらない！10のダメージ！
ばんばん★こ、今年はJaSST 行脚やめとこかなと思いつつやっぱり新潟に旅立つらしいw
おらかみ★暖かくなってきましたが、日々寒暖の差があるので、お気を付けてくださいね！
健康第一！！

★あくづけ★

2012/03/11(Mon)はっころ
WACATE-Magazine へんしゅろ
☑Magazine@wacate.jp
<http://wacate.jp/Magazine/>