

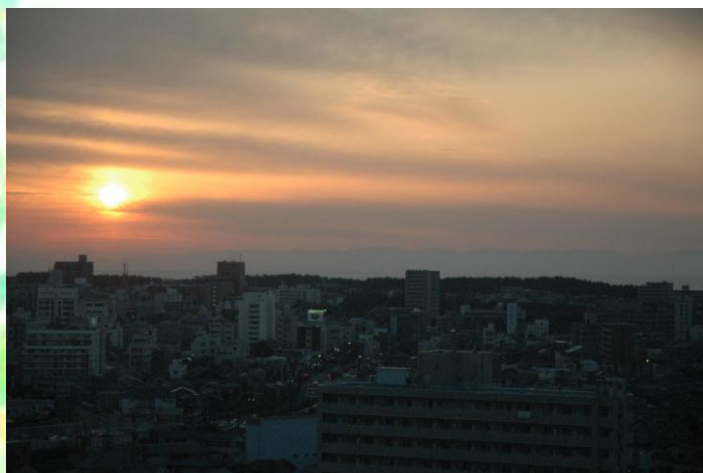
WACATE マガジン Magazine Vol.38

🌸ご挨拶🌸

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

春分を迎え、徐々にあたたかくなってきました。
花粉も飛んでおりますが、
日差しに春を感じるようになりましたね。

写真は JaSST' 12 Niigata の開催地
万代島ビルから見えた夕日。
うっすら見えるのは佐渡です。



とゆーワケで行って来ました JaSST' 12 Niigata !

今年も満員御礼となった大人気の Niigata。
話題が熱く、素晴らしいです。

がっつりレポートさせていただきます！

WACATE-Magazine vol. 38、
はじまるよー！

🌸お品書き🌸

【特集】

JaSST' 12 Niigata Report !

【おまけ】

にいがた酒の陣に行ってきた！

【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

【連載】

今月の STAR☆

【あとがき】

🌸WACATE Topics🌸

WACATE2012 夏 開催決定！
2012/06/30~07/01！！

WACATE2012 夏 テーマ決定！
詳しくは[Web](#)で！

JaSST'12 Niigata Report

—にいがたの底力—

とゆうワケで行ってきました。新潟は万代島！
今回も大人気で満員御礼の JaSST'12 Niigata に
WACATE-Magazine 編集部が潜入してきました！



■オープニング

実行委員長あいさつ



“JaSST 新潟の第1回目である昨年、基調講演をされたにしさんから、こんなことを言われました。

「新潟ならではのテスト品質を考えよう、他の地域が新潟のマネをするようになろう」

—第2回目である今年は、新潟の足場を固める場になりたいと思います。”

と、今回の JaSST'12 Niigata への熱い想いを告げました。

そして、基調講演のはじまりです。

■基調講演

XDDP による品質と生産性の同時達成

清水 吉男氏 (株式会社システムクリエイツ)

◇はじめに

今回は品質と生産性を両立する重要性和、それを実現する XDDP について解説する。

なお新潟については、AFFORDD(派生開発推進協議会。清水さんが設立・運営する派生開発についての団体)の会員がおります、XDDP の普及の糸口をどうするか考えていた。今回は非常に良い機会。



今回紹介する XDDP、USD、PFD は最近普及が進んでいる。例えば日本各地で最近勉強会が色々開催されている。また第五回 5WCSQ(世界ソフトウェア品質会議)で XDDP の論文が Best Paper Award 受賞。XDDP はジャパンローカルの方法論でなくなった。

◇派生開発について

この講演では派生開発を扱う。

派生開発の公的な定義としては JIS のものがある。JIS では保守開発を定義。それを訂正保守と適用保守といった更に細かい区分けで定義している。最近では改良保守や緊急保守が追加された。ただ派生開発と少し異なる上、色々な開発を一つの保守開発として扱っているため危うい。

世の中の開発の 90%以上は派生開発。キャリアの中で全く新規開発に関わることがない人もいる。

ただ注意が必要なのは世の中の開発方法は新規開発についてのもの。標準プロセスもそうであることが多い。派生開発にとって最適化されていないことがある。

◇派生開発の注意点

派生開発では注意点がいくつかある。

まず全体を理解することが困難であるという、**部分理解の制約**を受ける。ただそれを理解せず、「全体を理解していたら開発は失敗しなかった」という失敗の理由付けしてしまう現場を良く見る。全体を理解するのは困難だし、そもそも「全体」「理解」という言葉は定義できないし曖昧。そんな現実的でないことを理想にしつつ、部分的な理解の中で変更作業を進めることがよく見られる。

次に**稚拙な変更をしがち**。開発期間のプレッシャーから、変更箇所が目についたらすぐソースコードを変えるというスタイルが生まれている。そこでは開発初期の段階で、浅い理解のまま不適切な変更をしてしまうことになる。それが原因で問題が埋め込まれ、結果的に納期が遅れてしまうということが出てくる。

なおそこでは不適切な変更だったと後から気づいても、すでに多くの変更を重ねていることから、修正が困難なことも多い。

補足だが、**変更要求の実現手段は色々ある**。例えば画面の仕様変更にしても、View を変更するか、View の制御部を変更するか、View 制御に使われるモデルやロジックから変更するか、といった選択肢がある。稚拙な変更では、そこで適切な変更方法を選択するというのが困難。変更後のコードレビューでチェックするにしても、判別も難しいし、問題発見後の修正も難しい。

なお**稚拙な変更を続けると人格が壊れる**。顧客に取って後ろめたい変更を安易に選択するようになって、人格が蝕まれる。

◇品質と生産性について

ソフトウェア開発の利益工場主団として開発コストの削減がある。コスト引き下げ方法は二つ。生産性の向上で対応する方法と、オフショア/ニアショアによる削減。

後者は価格の叩きあいになりがち。品質と生産性でのダントツの差別化が難しく、選択肢が発注側になりがちなため。またそれ以外のコスト引き下げ競争でも、人件費の安い国にまず勝てない。

前者は日本でも戦える。生産性は利益を生む。品質が高くても生産性が低ければ利益がでない。**品質だけでなく生産性も良く見なければならぬ**。ただし前者の実現に**品質と生産性を同時に達成する開発プロセスの設計力**（清水さんはプロセスの最適化やカスタマイズ、テラリングを、プロセスの設計と呼称されている）が必要。また生産性のデータ収集も重要。

ただ現場では品質データ（レビュー指摘率、KLOC あたりのバグ発生率、仕様変更率など）は一般的に評価されているが、生産性のデータはあまり評価されていない。結果、生産性を犠牲にした改善が行われている。

なお、**品質と生産性は相反するという考え方があるが、そうでもない**。両方を考慮したプロセスを考えないといけない。片方のみだともう片方の犠牲が大きくなる。特に生産性は犠牲になりがち。



◇XDDP の背景

XDDP は生産性と品質の両立を要求されて作られたプロセス。1978 年にアメリカ主導の派生開発でプロトタイプを構築。それなりの規模で悪条件もあったが、短納期開発を実現。ここでは「変更前後の仕様をレビュー」「見積もりがしやすい、必要最小限の成果物とプロセスを構築」という工夫により、品質と生産性両方を確保した。その後、XDDP、USDМ、PFD（当初

は PFD）のトライアングルを考案。

USDМ は要求管理のためのツール。追加・変更の仕様漏れを削減し、XDDP を支える。

USDМ では仕様を「関係者が同じものをイメージできる状態の仕様」という意味付けで使用。また要求仕様書という成果物を定義。要求仕様書とは「開発で実現してほしいことについて、“開発する関係者”がその実現方法の認識を特定できている文章」。作るための必要最小限のドキュメント。

PFD は開発プロセスの設計のためのツール。無駄のない合理的な開発アプローチの設計や、状況変化に応じたプロセス調整を実現し、XDDP と USDМ を支える。

PFD ではモデリングによってプロセスの動きを明示化する。プロセスは本来変えていくものであり、例えば熟練度に応じて使用の粒度を調整したりする。PFD はそれに有効

この **XDDP、USDМ、PFD のトライアングルが生産性と品質を支える基本構成**。

◇XDDP の詳細

一般的な方法では、変更要求が来たら、変更箇所をみつけてすぐコードを変更するフローを繰り返していく。変更に変更を重ねて進める。

一方 XDDP では、「要求仕様書で変更要求を具体化」→「レビューで要求仕様書が適切かチェック」→「要求仕様書から変更方法を検討し、設計変更書に書きだす」→「レビューで設計変更書が適切かチェック」→「実際に変更を実施」の流れ。これにより不適切なコード変更の防止や、コード変更前の問題除去、プロセスの問題改善を促進める。

なお XDDP では追加機能要求仕様書と、変更要求仕様書で要求を管理する。

◇追加機能要求仕様書

追加機能要求仕様書は、新規開発の要求仕様書と似ている。関係者が要求に対する機能の内容を特定するための文章。

追加機能要求仕様書では網羅的な仕様化で仕様漏れを防ぐ。仕様化の方法としては、適切に記述された要求に含まれる**動詞を単位に仕様を展開することを実現する**。例えば「計測データを受信し、平均値を算出しながらリアルタイムに表示する」という要求なら、「受信」「算出」「表示」それぞれの仕様について詳細化する。

また状態遷移では、遷移の際のアクションについて仕様を出し詳細化する。

なおユースケースでは、詳細な仕様化が困難。ユースケース分析の途中で XDDP の手法に切り替えて詳細化する。

◇要求仕様の分割・階層化

要求は分割・階層化する。一つの要求に動詞の数は 7 つぐらいが良い。それ以上だと分割・階層化を進める。分割・階層化を進めることで、仕様漏れが見えてくる。

仕様の抽出に当たっては、動詞と目的語を意識した仕様のグルーピングも重要。適切にグループ分けし、グループごとに仕様化することで、仕様の発散を抑える。

例として、画面を扱う場合は、画面要素ごとに要求を書き、要求の動詞や目的語から仕様のグループを抽出。そしてグループごとに仕様化する。

仕様の記述レベルは、設計担当者は実現可能性を把握でき、またテスト担当者は検証可能性を把握できるまで書かなければならない。S

なお上記のような工夫は、適切な日本語で適切に要求や仕様書が書かれていることが大前提。

また仕様変更率は5%以下が望ましい。変更仕様数/総仕様数 * 100 が5以下であれば開発が判定する。対応も用意。最善は3%。仕様化の際も留意すること

◇変更要求仕様書と設計との関係

変更要求仕様書は作るための文章。関係者が変更内容を把握できるまで書く。例えばソースコードレベルまで変更内容を記述する。

変更要求仕様書では before/after を明確化する。世の中では after しか書いていないものが多いので注意。

また変更要求仕様書でも、前述したような動詞と目的語への着眼やグルーピングを使った仕様化を行う。なお追加機能変更仕様書と、変更要求仕様書は対応付けが必要。前者が書かれれば、その実現方法を後者に明記する。

変更要求仕様書を作成したら、モジュールと変更要求を関連付けるトレーサビリティマトリクスを作成する。トレーサビリティマトリクスが変更要求と変更設計のつながりを管理。

変更設計書では、ソースコードの変更方法と変更時のテストを作成。派生開発では変更箇所にはバグが出るので、ここでのテストは注意して行うこと。

まとめると、**変更要求仕様書で What/Why を書き、トレーサビリティマトリクスで Where を書き、変更設計書で How を書く。それらが問題ないことをレビューで確認してから、一気にソースコードを変更する。**駆け抜けは禁止。

なお一般的な機能仕様書や設計仕様書は、XDDP ではテスト後に変更要求書や変更設計書を旧版にマージすることで作成。

◇保守性は大事

保守性は競争力につながる。ソフトウェアの保守性は分解しても分からないので模倣ができない。

そのため保守性も変更要求に加えよう。例えば「現状の保守性のレベルを損なわない」というような要求を追加しよう

◇品質と XDDP について

開発を川に、そこでのバグを川に流れるゴミとえらるなら、テストは川に流れるゴミを見つける作業。テストを強化するというのは、網を細かくしたり、網の数を増やしたりすることと位置づけられる。

なお品質とは、川の水質のようなもの。網とは関係ないが、テストの結果で水質を推測することができる。

ソフトウェア開発というのは、変換プロセスと検証プロセスを合わせたもの。変換プロセスが合理的であれば、検証プロセスがうまく動き、品質保証の基盤となる。XDDP はそこをうまくやっている。

なお生産性データは注意が必要。例えば全行程工数に対する生成行数、実装工程工数に対する生成行数を比較してみる。差が大きいなら実装前の工程が整備されている可能性がある。XDDP ではその差がとても大きくなる。

◇まとめ

近年、ソフト開発力はとても重要。製品の差別化にソフトウェアは使えるし、工場と違って国内でも競争力を発揮し得る。それゆえ**ソフト開発を続けていくなれば、品質と生産性の両立は重要。**

なお XDDP は品質とともに生産性も高める。生産性3倍・5倍も可能。また XDDP は色々な取り組みの基礎になる。例えば SPLE がそうで、コラボレーションした方法論が検討されている。

◇質疑応答

●デバッグのコード変更はどう管理するか
→バグ修正についても XDDP で管理。デバッグの変更要求仕様書を発行し、レビューして、変更設計書を発行し、レビューして、コードを修正する

●要求を決められない時は
→XDDP を二段階にする。追加で変更要求がでたら、また変更要求書を発行し、そのまま XDDP の方法へ

◇セッション感想

清水さんの提唱されている XDDP、USD、PFD は何十年も効果を発揮し続けている方法や考え方です（ソフトウェア開発業界が劇的なスピードで変わり続けているのを考慮すると、これは驚異的です）。それだけに、洗練されながらかつ実践的な示唆や学びに富むセッションでした。

なお清水さんの方法や考え方は、極論すると「ソフトウェアをトラブルなく効率的に開発するために必要な最小限の事」を、隙なく堅実に実践するものです。それだけに、**私達は本当に必要な事を喧騒の中で度々見失ってきたという現実と、大事なことを堅実に実行していくことの重要性を痛感させられました。**



休憩時には JaSST 10 周年記念誌の紹介がありました。

■『NIIGATA SKY PROJECT』でのプロジェクト推進ー短期・分散開発での確実なジェットエンジン開発ー
岩田 拓也 氏 (産業技術総合研究所)



◇NIIGATA SKY PROJECT について

新潟市の産業支援の対象に、下記4つがある。

- ・航空機・自動車等、機械・金属関連産業
- ・組込み・高度 IT システム関連産業
- ・食品・バイオ関連産業
- ・港の活性化につながるつながる産業

産業支援に航空機産業を指定しているのは、新潟市、中京地区、栃木圏域の3地区のみ。

NIIGATA SKY PROJECT では、現在の航空産業の支援 (JISQ9100 や NADCAP の認証の取得)、および別の (将来の) 航空産業への挑戦を支援している。

◇なぜ新潟ジェットエンジンなのか？

上記のとおり新潟では、航空産業振興を行なっているが、現在の産業は、部品産業の後押し。

日本は後発組、下請けの国。

これから、日本発、そのなかでも難しいものを新潟発としたい。

二次の時代に来る、別の航空産業を目指す。

「将来の別の航空産業」とは・・・

「おばあちゃんを乗せたら自動的に孫に会いに運んでくれる」ような航空機

トラフィックシステムの実現。点と点を結ぶような運行ができたら・・・

→PAV (パーソナルエアビークル) の実現を目指している。

2025 年に PAV 運行管理システム運用開始予定となっている。

ネットワーク制御を行うことで航空機を運行。運転は自動で乗客がいる、有人無人機。

今の航空機は速さを求められるが、陸上の交通機関はそうでないものを求められている。電車があるのに、電車より遅い自動車を利用する。自動車なんて遅くても買う。200万でも買う。

そういったもの→「遅く飛ぶもの」が空にはない。

固定電話が携帯電話になったように、空中移動も個人を対象としたものにする。空港を利用するのではなく、ドア to ドア交通機関にする。そのためには、低速で安定した飛行体が必要。

また、そのために大事なものは「免許が不要なこと」。運ぶことを専任するところを設ける。

そのためにはプロセスが必要。まずは無人で物を運べることを考える。そのうち人を乗せて運べるようになる。

無人航空機飛行特地区として、「アジアのドーバー海峡」佐渡海峡を利用する。

使用した技術は、静音ジェットエンジン技術、飛行機械技術など。

重力を利用することで、重くても舵がとれるようにする。

実演の動画紹介。

なるべく短距離で離陸→螺旋状にあがっていく。そういう研究をしている。

空中輸送ロボットに使用する動力の、絶対条件は「安全」。従って、プロペラはダメ。ファンも小径化が必須。

ジェットエンジンを利用するメリットは、

- ・露出する回転部分が無い (=安全)
- ・回転運動なため、高信頼性があり低振動
- ・小型軽量高出力

デメリットは、

- ・騒音
- ・燃費

そこで、騒音低減と燃費低減を目標とし、達成方法を以下とした。

- 1) 従来比燃費 20%以上の向上→従来比 20%の軽量化
- 2) 重量あたり推力比 20%向上→従来比 20%の軽量化
- 3) 従来比 15DB 以上の騒音低減→静音装置の導入
- 4) エンジン保証時間の 50%向上→寿命上決定点の特定と改善

金属加工技術や制御技術を駆使して取り組んでいる。

アセンブリされたものをどうやってソフトウェアで動かすかが課題。これには非常に高い技術を要する。

短い期間での開発を実現すべく、新潟地域にジェットエンジン開発に必要な設備を整備した拠点を設けた。プロジェクトに関わった各社の所在地は散らばっているが、「ここにすれば進捗がわかる、状況がわかる。」そういう場になった。

「航空機関連産業と組込みシステム関連産業をうまく組み合わせることで、短期開発ができるであろう。」として、プロジェクト設計をした。

→目標を達成した。わずか7ヶ月でのジェットエンジン開発に成功した。

この先、大きな目標に向かってチャレンジしていきたい。

◇セッション感想

講演の中では、動画で実際の飛行実験の様子が紹介されました。

ハード面でもソフト面でも、日本が得意としている技術を盛り込んでいるのを感じました。

今後の展開が楽しみです。

■ 「NIIGATA SKY PROJECT」での

ファームウェア開発の取り組み 吉田誠氏 (NEC ソフト株式会社)



◇NEC ソフト株式会社について

新木場に本社があり、システム構築、保守・運用、アウトソーシングの IT サービスを提供するなどの SI 事業、各種ソフトウェア開発をしている。新潟支社の 6SI グループでは、スマートフォンを中心とした組み込みソフト開発をしている。

◇NIIGATA SKY PROJECT のソフトウェアの概要

経済産業省の採択事業としてスタートした。NEC ソフトは以下のソフトウェア部分を担当。

- ・ボード上で動く小型ジェットエンジン制御
- ・PC 上で動くジェットエンジン状態モニタ

開発環境は、運用支援ツールにサイボウズ Live、品質管理ツールに NEC ソフトの標準である Progress901 など、無償のツールを使って、開発費用を抑えている。

プロジェクトの特徴として、ソフトウェア開発は 5 ヶ月。ハードウェアもソフトウェアもゼロからのスタートで、目標はエンジンをまわすこと！明確な要求仕様はないが、似たようなものとしてオランダ製の小型エンジンを参考に、走りながら仕様を固めていく進め方だった。

◇ソフトウェア開発にあたって

ジェットエンジンという馴染みのないソフトウェアが開発対象なので、非常に懸念事項・不安が多くあった。

懸念点①：ジェットエンジン、メカ制御の経験がない

懸念点②：要求仕様がない。何が正解かわからない。

今回は研究開発なので、試行錯誤しながら決めていくところに時間を費やした。

懸念点③：設計とテストをどう行うか。

どのように設計を行うか、プラットフォームも出来上がっていないところに、どのようにテストを行っていくかに不安があった。

懸念点④：品質指標をどう設定するか

前例がないソフトウェア開発なので、品質をどのように考えて開発をすればよいのかという懸念があった。

とにかく、ジェットエンジンという、未知の世界のソフトウェア開発だったので、どういうふうに取り組んでいくか、というところが共通の課題だった。

◇懸念事項に対する施策

施策①：パートナー会社から経験者を投入した。また、社内教育で、ノウハウを吸収するようにした。

施策②：基盤等ハードウェアの担当している企業と、早期に、また随時に意識合わせをし、仕様を詰めるようにした。具体的な部分はオランダ製の小型エンジンの動作・仕様をたたき台に、少しずつ固めていくやり方をした。

施策③：研究開発なので、変更・修正が入ることを前提とした設計をし、ソフトウェア構造に工夫をした。テストは、ハードウェアの方も新規開発のため、シミュレータが必要となり、シミュレータを独自開発した。

施策④：従来のプロジェクトである携帯電話よりも相当高い目標をつけなければならないと考えた。

◇開発方針3つ

懸念点と施策を考慮し、開発方針を3つ設定した。

①ソフトウェア開発プロセスにおける活動について

携帯電話の開発でも品質は厳しく言われるため、品質向上活動をしていた。その活動を参考にして、プロジェクト管理ツールを導入し、各工程のバグ数や内容、どこにバグが多いかを分析した。

②各モジュールの独立性を高めた

通常の開発でもやることだが、要求仕様に変更が多く入るということで、重要だと考えた。

安全面に重きがあるので、一つのモジュールで問題がでたら、別のモジュールで検出できるように考えた。

③異常検収時のソフト制御規定をつくった

◇開発における取り組み～品質

レビュー時間は携帯電話の開発の 1.4 倍くらいを費やすようにした。バグの数は単体テストで 3.5 倍の検出を目標にした。実績としては、バグ数が工程ごとにバラツキが発生したが、トータルでは目標を達成している。シミュレータを同時開発しており、コードレビューより単体テストでシミュレータによって多くのバグを検出したほうが、効率が良いと考えたことが、バラツキの理由の一つである。

◇開発における取り組み～ソフトウェア構造

ハードウェアにあわせて柔軟に対応できるように、モジュールを階層化した。仕様の変更への対応だけでなく、デグレードし難い作りになったと考えている。また、ハードウェアが変わってもドライバ層だけ変更し、AP 層はそのまま使えるため、移植がしやすいという利点がある。

◇開発における取り組み～PC ツール

PC ツールとは、ジェットエンジンの状態モニタや調整のソフトウェアのこと。PC ツールの機能には二つの役割がある。

- ・状態をモニタリングする機能
- ・ジェットエンジンの制御をするための閾値を自由に設定する機能

閾値は 38 種類出力できるようにしている。主に異常値、リトライの閾値を様々に設定できる。

◇開発における取り組み～シミュレータ

ジェットエンジンの制御システムをそのままシミュレートしたものを作成した。

目的はAP層の単体テストの環境としてだが、PCツールの動作確認にも役に立った。また、結合テストの工程を待たずにAP層のバグを抽出することができて、結果、結合テストではAP層のバグは1件も出なかった。

◇開発における取り組み～エラー制御

ジェットエンジンの制御は、OFF>STANDBY>STARTUP>RUN>POWERDOWN>OFFの状態遷移が起こる。

STARTUPシーケンスにおけるエラー制御では、ガスの着火に失敗すると、ガスの着火条件を自動補正し、ガスの着火をリトライする。リトライが一定回数を超えると、エラーと判断する。

RUNシーケンスでは、温度異常、回転数異常など、設定された閾値で自動的に補正するようになっている。閾値はPCツールで自由に設定できるようになっている。

各種センサーの異常を感知すると、POWERDOWNシーケンスに移行する。

スイッチが変更された場合は、ユーザによる操作かどうかを判定するため、1秒後に再度判定をすることで、ユーザ操作の判定をする。

◇成果～問題から学んだ教訓

ROM（本番環境）とRAM（開発・試験環境）で動作に差分が発生した。これはROMの実行速度が早く、割り込みの回数などで判定していた条件の見直しが発生した。

加速度/ジャイロセンサでは、動作が不安定になることがあった。これは試行錯誤をし、wait処理を入れることで回避した。

これらの事象は、想定できたことかもしれないが、経験不足から影響の度合いが想定できなかった。早くから一部でもROMでの動作確認をすればよかった。

回転数の取得で問題が発生した。当初に設計していた方法では回転数があがっていくと、正常に取得ができないため、取得方法を見直した。階層構造、機能分割をしていたので、その部分のみの対応で済んだ。

燃焼実験でエンジンが炎上した。PID制御（燃料ポンプの制御の値を計算する）で想定以上に供給量が多くなってしまった。POWERDOWNで燃料の吐き出しをしたが、さらに炎上した。

シミュレータでテストをしていたが、本番環境では想定以上に回転数が遅く、燃料を吐き出し過ぎたことが原因。（ここで炎上の瞬間のビデオが流れました。炎上と言うか、大噴出！）新潟市のプロモーションビデオには使えない映像かなと思っている。

◇成果～得たもの

PID制御理論の習得。

これまで経験のなかったことを理論的な習得としてできたことと、ソフトウェア部品の資産が成果となっている。

ジェットエンジンの制御。

安全制御の面について、非常に大きな成果となったと考えている。また、開発経験ができたことも成果である。

航空産業の動向の理解ができた。今後のビジネス検討材料にもなると考えている。

人脈。

NECの関係会社とのつながりだけでなく、産業技術総合研究所を始めとした新潟県の各企業とのつながりができた。

◇今後の課題

ソフトウェア制御可能な安全制御を追加する。

燃料供給タイミング、係数の最適値を検討する、回転数上昇の評価のタイミングなどの改善。

PCツールの設定画面拡張、エラーを画面に表示する。

その他、エンジン制御のノウハウを、どのように活用していくかも課題と考えている。

◇最後に。。

吉田様は途中で別プロジェクト（東京）にアサインされ、実際にエンジンが動くところを見れなかった。東京へ行ってから、東京国際航空宇宙産業展2011へ行き、そこで新潟市の講演とエンジンの実験があり、初めてエンジンが動くところを見て、非常に感動した！

当時の担当者からやりがいがあった、楽しかったという話を聞いた。決められたものを作るだけでなく、仕様、設計やテストで試行錯誤を繰り返していくという、モノづくりの原点を体験したプロジェクトである。

◇セッション感想

経験がないということからリスクをしっかりと洗い出し、対応していくという、プロジェクトの進め方の基本のお話と、「動いた！」という感動を持つモノづくりの原点というお話でした。動いて当たり前と思われている昨今、エンジニアが作ったものに感動できるというのは素晴らしいと思いました。



■システム自動生成ツール「GeneXus」を用いた
ユーザー視点開発
山内 啓悦氏(株式会社ウィング)



基本的には Java C#が主流かと思いますが、弊社は GeneXus を使っています。
ここで会場の皆さんに質問。GeneXus をご存知ですか？(ちらほらと手があがる)
…馴染みは少ないですね。
とはいえ、首都圏では割と事例も出てきています。

◇GeneXus とは。

ウルグアイで開発された DOA による自動化ツール。
データ分析結果と業務ルールを入力することによりシステムを自動生成する。
アメリカでは評価が高いマルチ言語対応の統合開発ツール。
Java, C#, Ruby など扱える。
次はモバイル端末向けに出るらしい。
Oracle, SQL Server, MySQL, Postgres など扱える。
同じようなものはあるが、上記のような「マルチランゲージは」珍しい。
日進月歩の IT 技術は GeneXus に任せて技術者は業務仕様把握に努めることができる。
—GeneXus は 20 年以上の歴史を持つそうです。

◇GeneXus のメリット

資産の継承。ビジネス部分とテクノロジーを分離。

・画面、業務ルール、データ構造を基本構造とする。
GeneXus はデータ志向なので、まずはデータを投入する。それをどうやって出力結果にするか、を自動生成する。

・ナレッジベースでできる。

GeneXus8 で作ったサイトを GeneXus10 でリプレースするのが2ヶ月で終了できる。
サーバサイドの動きはあまり変わらず、活かすことができる。

書籍もあるので、見てみてください。
日経コンピュータの方でも特集を組まれたので、見ていただければ、と。

ドールさんや鈴廣かおこさんなどの事例が掲載されているそうです。

◇GST (GeneXus System Template)

ある程度テンプレートがあり、最初からシステムを見える化する。最初は WF で実施していたがどうしてもうまくいかなかった。

プロトタイププロジェクトで3例ほどあります。
東京農工大学さん、某H社さん、某N社さんで一番良かった数値でJavaの2.5倍の生産性が出せる。
※ただし、GeneXus に向いているものが前提です。
なんでもかんでも生産性が出るわけではない。

早期のデータモデルの確定が重要。
要件定義の最終成果物は DFD。

機能の”プロトタイプ”というよりも、”ミニシステム”を作る。通常で言うと基本設計。
ユーザインターフェースなどを作り、プロトタイプを作る。
※ここで基盤を作る。

機能一覧はプロセスから起こす。
ERD はデータ志向で起こす。
DOA の世界だと、ボトムアップアプローチをして第三正規形にします。

乖離が発生するので、どの機能とどのデータが、というリレーションを明確化するので DFD を使う
結合テストのところでどのようなデータが流れるか、を見れる。本当に思っているとおりに流れるのかどうか確認できる。
※これらを要件定義で実施する。

◇ST-REAM (System Template
- Rapid and Easy Architecture Methodology)

・方式設計
非機能要件、UI を決めていく。

・プロトタイプ開発
並行作業でシステムの7%くらいの規模のプロトタイプを作成する。

要件定義と方式設計と並行しながら
PD フェーズ→RAD フェーズとフェーズを移行する。

OPD (Prototype Development) フェーズ：
-UI (User Interface) 設計 (プロトタイプ)
-BL (Business Logic) 設計 (プロトタイプ)
-PCL (Prototype Checklist)

ORAD (RapidApplicationDevelopment) フェーズ:

- UI 設計
- BL 設計
- 製造
- UnitCheckList
- 単体試験
- ユーザーレビュー

RAD フェーズでは設計・開発・試験を繰り返し実施する。基本、SEしかいない。

通常、プログラマを投入しない。SE=GeneXus 技術者。ただし、テクニカルな部分を担当する技術者はいた方がよい。



◇自動化ツール利用における品質

○GeneXus からの品質考察

良い点

- ・ 統合開発環境としての側面があるので管理における **デグレードが発生しにくい**
- ・ **自動化されている箇所においてはテストをする必要がない。**
- ・ データモデルとサーバサイドアプリケーション、クライアントが密接につながっているため、それぞれの IF テストを行う必要がない。
- ・ クロスタイプスクリプティング、SQL インジェクションに製品が対応している。
- ・ **AJAX 対応も自動的に行われているため、AJAX 対応も不要。**

注意点

- ・ GeneXus そのものに不具合があった場合に、対応に時間がかかる。
- ・ **ホワイトボックステストを行うことができない。**
- ・ DOA に沿っていないデータモデルを AsIs で利用し、アプリケーション側だけをリプレイスする場合は、開発は可能ではあるが生産性は落ちる。

○GST からの品質考察

良い点

- ・ 画面インターフェースの標準化を容易に行うことができる
- ・ コーディング規約を意識しないで開発できる。
- ・ すでに品質検証済である。各種処理部品を利用できる
- ・ 特に画面による振る舞いについても事前にユーザーレビューし、認識合わせができる

注意点

- ・ GST の仕様からかけ離れたもの、**過度なリッチクライアントな挙動を求められた場合、効果が半減する**

OST-REAM からの品質考察

良い点

- ・ ユーザーレビュー主体で開発をすすめるため、爆発はしにくい。
- ・ **ユーザに単体テスト完了機能について早い段階で確認作業をおこなってもらうため早期の品質向上が見込める**
- ・ 各画面での仕様統一を行いやすい

注意点

- ・ スケジュール管理が煩雑になる。
- ・ SE としての技能も必要となる
→ある程度の力が必要。ただプログラミングしていればよい、というものではない。
GeneXus のスペシャリストは必要と考えている。

◇質疑応答

Q. 開発規模が人月だったのですが、実際のコード規模は？

→FP で見積もりをしています。

画面数であってしまいますと、40 人月のものは 160 機能。FP 上の機能数になります。

※1 機能=1 画面。

200 ちょいくらいの案件もありました。

Q. GeneXus を使用した際、異なる際のコスト比較は？

→各工程ごとの工数削減についてですが

設計と製造が別れていないので、出すのが難しいですが製造ですと一番いいときで Java の 2.5 倍の削減。

設計ですと、データモデルのでき次第で変わります。※GeneXus 向けの設計になります。

テストフェーズですと、感覚ですが、単体テストは半分くらいになると思います。結合テストは 30% くらい削減。

Q. 自動化したときの DB のパフォーマンスはどうか。

→第三正規形でやっていると、あまり変わらないですが SQL でゴリゴリ書いていると、やはり難しい(あまりよくない)ですね。

—正規形を崩す場合のチューニングは？

→正規形崩しをデータモデリングをします。

→ERD まで戻ってやる？

→早くしなければならぬ点については View を使ったりして変則的にしたり、GeneXus の中でインデックスを使ったりします。

◇感想

自動コード生成ツールと事例の一部の紹介、そして活用方法について非常に興味深い内容でした。

自動で品質の高いコードが生み出されるという技術についてはとても魅力的であり、長い歴史を経て洗練されてきているのはとても素晴らしいことだと思います。ただ、使う際には入念な準備と検討が必要であり、特に向いている/向いていないについての見極めが重要となりそうだと思います。

また、保守については賛否両論な部分があるかな、という印象を受けました。

柔軟に対応するとなると、どうしても手を入れなければならないと思いますし自動生成されたものとの連携を確実に行うコードを書くのはとても困難と予測できます。その際に自動生成されたコードを変更する必要も出てくるでしょう。

しかし、その自動生成されたコードの仕様はどうなっているのか、などと考えると一筋縄ではいかなそうです。

なので、**検討・アーキテクチャ設計などの検討をGeneXusでやりやすいように最初に入念に検討をする必要があり、そうすることで非常に強力なツールになるのだな、**という印象を受けました。

◆クロージング



池田実行委員長からのご挨拶。

がっつりと品質やテストの話聴き、学んだことがあったと思う。

新潟の底力を感じた。紹介できてよかった。

2年目、今回も大勢の人に参加していただいた。

定着してきたかな、と思う。

1年に一度のイベントではなく、持ち帰り、フィードバックして新潟全体がよくなり、参加者のみなさんが登壇者としていらっしゃる日がくることを楽しみにしています。

ずっと続いて、盛り上がっていったら良いと思います。

昨年同様、実行委員長は感極まっております。

三浦さんの投影した夕焼けと相まって、一日が終わったのだ、と思いました。

■情報交換会 そしてLT

情報交換会は、新潟市中央公民館に場所を変えて開催。参加者にペットボトルドリンクとお菓子が振舞われ、前半は講師と参加者が自由に交流をはかる時間となりました。



後半はLT大会。

今回は5名のエントリーがありました。

まずは場をあたためるべく、実行委員の酒井さん登場！！

◇さかむきん さん



昨年 JSTQB 受けた→受かった！

しかし、JSTQB では、馴染みの無い用語がたくさん。

→**擬人化してみよう！**

という内容でした。

[初級編]

受け入れテスト=秋元康

ユニットテスト=チームA、チームK、チームB

統合テスト=AKB48

システムテスト=AKB、SKE、NMB・・・各ユニット

[中級編]

ホワイトボックステスト=オセロ松嶋

ブラックボックステスト=オセロ中島

グレーボックステスト=謎の占い師

・・・と、こんな感じで、

状態遷移図、デシジョンテーブル、境界値テスト、同値分割

オールペアアルゴリズムテスト、アドホックテスト、モンキーテスト、ロードテスト、探索的テスト
・・・に対し、擬人化していきましたw

だいぶ会場も和んだところで?! w次の方へw

◇笠井宏さん 「タスク管理 Redmine を使って」



Redmine を某スポーツのプロジェクトにつかってみた。という事例発表w

2008 年から。アジャイル形式で。
5月に一度大会、2月に本番!

システム構成はクラウドサービスで
プロジェクト環境は各拠点で分散開発
クラウドサーバに Redmine を入れる

どのように使ったか
基本: 1タスク1チケット
進捗はロードマップで。ロードマップがリリース計画に相当。
バージョン単位がイテレーション。
ロードマップの Ver. 単位で残数確認。
チケットの確認はネットから♪
議事録は Wiki を利用。
大会期間中は時系列の状況を掲示板で行う。

バージョン管理も行う。
チケットと関連付けが間違えたらコメント記述。
開催地(拠点)が違って大丈夫♪

ツールを勉強するときには、ソフトウェア開発に限らず、適用できる場を考える。
こういった考え方はとても有効だと思います。

次っ!
WACATE 編集部から2名エントリーさせて頂きました^^

◇編集部員ばんばん 「派遣テストエンジニアの想い」



みなさん楽しいネタを持ち込んでいらっしゃる中、空気を読まず一人マジネタを持ち込みました^^;

一般派遣でテスト業務に関わっている人はたくさんいます。未経験からテストを行う方も多いです。その中で、勉強したくてもきっかけがなかったり、勉強できることを知らずに過ごしている人も少なくはありません。そして、多くの派遣会社には、テストの勉強を支援する仕組みがありません。現場で勝手に育つことを期待しているのです。なので、勉強しようという意欲のある派遣スタッフさんを見つけたら、是非きっかけを与えていただけませんか?

というメッセージを送りました。

◇編集部特派員(WACATE 実行委員)なかさや 「WACATE スタイルで話そう☆てすバナかたるっcha」



・・・はい。表題のとおりですw

JaSST'12 Tokyo で行った WACATE セッションの紹介と、WACATE とはなんぞや? というお話、そして、本 Magazine の企画、てすバナかたるっcha の紹介。

てすバナかたるっcha は、オンラインなので新潟からでも参加できますよ♪
是非、新潟から発信しませんか?

というお話をしました。

◇実行委員長池田加奈子さん

「俺、このJaSST新潟が終わったら結婚するんだ…」



バイオリンとともに登場ですよ！！

結婚して埼玉県に移ります・・・

大変だ、新潟の実行委員が一人減る！

ということで、**新潟の実行委員募集！！**

こんなメリットがありますよ

- ・登壇者に呼びたい人を呼べる
(呼べるかどうかは相手次第けど)
- ・日本中のテスト業界の有名人と知り合いになれる
(でもって実行委員長とか振られる、
いやいや機会を与えていただける)
- ・休日や昼休みの間に効率的にタスクをこなす
スキルが身に付く
(昼休みに銀行のロビーで作業しているとかw)

やる気と熱意があってキリキリ働いてくれる人大募集！ww

w

残り時間で、バイオリン演奏を披露していただきました♪



◇乱入：JaSST 書籍販売担当？！長谷川聡さん

このテストの年表、すごいよー！今なら本がついてきます♪
(ちーがーう-www)
(しかし、年表も素晴らしいのは事実です！！)

JaSST 記念誌つくりました。

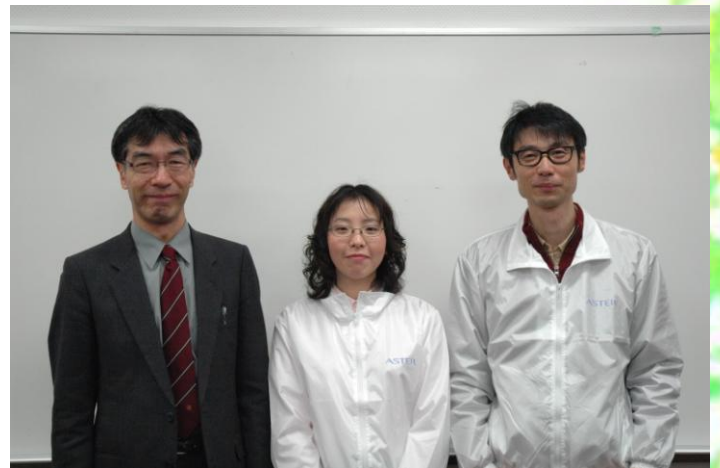
創立の話など、もりだくさんですよ。
JaSST でしか買えません！

なので、、次は JaSST 四国の会場へみんな Go！！



最後は JaSST で恒例の書籍 Get のためのじゃんけん大会！
Magazine 編集部からも同人誌 ManiaX 全 6 冊寄贈させていただきました♪

そして最後に実行委員の皆さんの記念撮影！



ーやりきった。そう感じられるいい笑顔ですね。

濃い一日を、有難うございました。
実行委員長の気持ち、実行委員の気持ちが表れた、
とてもよいシンポジウムでした。

ありがとうございました。

書いた人：WACATE-Magazine 編集部

行ってきました! にいがた酒の陣!

テストの勉強をした次の日ー



朱鷺メッセに人が押し寄せたー

今年は去年話題になった「にいがた酒の陣」に
JaSST' 12 Niigata が時間を合わせたということ
で、さっそく行ってみた!

そこで待っていたのは…
すっつっつっつごい人! w



中に入るのに結構並びました。そして中は…
すっつっつっつごい人! w



この人数がほとんど酔っぱらいになると思うと
…ゲフンゲフン。

中には会場特設の試飲コーナーなどの他、88の
酒蔵が集まり、さらに新潟の名物を食べさせてく
れる屋台など…たくさん新潟の名物・美味しい
ものが集まっていました!

試飲コーナー



酒蔵さんの試飲コーナー



とにかく凄い人！@@



屋台ではもちぶたをはじめに焼き串や焼き物、おでん、汁物、うどんなど人気でした。



肉をくらう肉々しい人。



いや、ホント食べ物もお酒もとっても美味しくて
ついつい色々なものを食べてしまいました。

コシヒカリのいなり寿司やおにぎり、
米粉を使った「にいがた焼き」やシュークリーム
やら…それはもう色々w

去年の JaSST' 12 Niigata の次の日は
月岡温泉で飲み比べをしましたが、
今年はさらにたくさんの飲み比べができました。

そして清水さんを囲んで飲みながらお話♪
清水さんも新潟名物にとっても喜んでいました。



他にも沢山の酒蔵の出展がありました。
中には 1 本 5000 円以上するお酒も試飲できたり
とってもお得！



なかには 30 もの酒蔵を回った強者もいたとかい
ないとか…w

かくいうワタクシは 10 くらいの酒蔵を回り、
そのうち 1 つの酒蔵に何度も行って何度も試飲
してさらには 2 本ほど買って帰るというダメお
やじぶりを発揮してまいりました。

いや、だって辛口おいしいんですもの。
仕方ない。

こういった大きなイベントだと思っていなかつ
たので、とても楽しめました。

JaSST などのイベントが、こういった地域の特別
なイベントに近かったりすると、その地域の素晴
らしいところも味わえて一石二鳥ですね♪
来年も楽しみです♪

書いた人：コヤマン

Software Test Topics

当番：いのみん (Magazine 編集部)

こんにちは。すね…いのみんです。

JaSST' 12Niigata も終わり、翌日の酒の陣でへべれけになった方も多いはずw
一息つくかと思いきや、WACATE2012 夏の日程も告知されて、6月に向けてメキメキとスキルを上げるべく、日々勉強をしている方も多いことでしょう。うんうんw
4月5月は東海や関西の勉強会がアツイみたいですね。参加された方は是非 Magazine に投稿して貰えると嬉しいな♪ (てへっ)

【勉強会】東海合宿

<http://kokucheese.com/event/index/27280/>

日時：2012年4月21日(土曜日)13:00
(12:00 受付開始)
～2012年4月22日(日曜日)17:00

場所：ヤマハリゾートつま恋

内容：

- ・ライトニングトーク (TL) による自己紹介
- ・ワールドカフェ方式(※)によるダイアログやディスカッション
- ・お酒を交えた語り合い
- ・SIG
- ・ふりかえり

参加費：¥15,000～¥20,000 (宿泊費※)
+ 雑費 (備品代の割り勘)

定員：20名(予定)

WACATE 合宿東海版? WACATE2011 冬で盛り上がったワールドカフェを再び!!
東海のアツイ夜は春の嵐を巻き起こす!?
WACATE2011 冬に参加できなかった方や、まだ足りない「もう一杯! おかわり!」な方は是非!

【勉強会】実践的! なぜなぜ分析の活用テクニック 【実習付き】

<http://atnd.org/events/25908>

日時：2012/04/27 10:30 to 16:30

定員：30人

会場：TIME24ビル 4F セミナールーム

URL：

<http://www.rdsc.co.jp/seminar/120429.html>

なぜなぜ分析シートを使ったグループ演習あり! 変更管理や品質管理についても勘所などが紹介される?…かもしれない。

【勉強会】第2回 Android テスト祭り

<http://atnd.org/events/26800>

日時：2012/04/28 12:30 to 18:00

定員：300人

場所：東海大学 高輪キャンパス

参加費：無料

前回好評だった Android テスト祭り第2弾!!
第1回目の70人枠を約3倍に拡大しての臨戦体制でお出迎え♪
既に200人もの参加表明がされているところを見ると、注目度はなまる上昇中!

【勉強会】WARAI (関西ソフトウェアテスト勉強会)

<http://kokucheese.com/event/index/33351/>

日時：2012年05月12日(13:30～17:30)

場所：デジタル・インフォメーション・テクノロジー株式会社

(大阪府大阪市西区江戸堀1-5-16 肥後橋MIDビル3F)

参加費：無料

定員：25人(先着順)

申し込み終了 2012年05月12日 13時30分まで

今回は「ソフトウェアテスト技法ドリルーテスト設計の考え方と実際」の書籍をもとにグループ討論やワークショップになります。
35歳以下を対象としたテストの基礎、テスト技術について身体で覚える、関西のソフトウェアテストエンジニア主催の勉強会♪
WACATEとは関係ありません!(ｷｯ)

このコーナーでは、書籍やイベント・勉強会の情報を随時募集しています。

「来月勉強会開催します! 参加者募集~!」や「こんなイベントあるけど、どう?」「開発系イベントだけどテストの話とコラボするので、ちょっと参加してみない?」等でもOKです。
開催地も日本全国どこでもOKですので、貴方の知っているとっておきな情報をお待ちしております(^ω^)/

今月のSTAR★

◆肉食系スター☆

WACATE 実行委員会が肅々と終了したのち、皆でジンギスカンを食べに行くことに！

普段から頑張っている実行委員たちに、スター☆自らジンギスカンを作ります。

見よ、この手さばき！早すぎて映像がかすみます。



そして取り分けるのもこの手さばき！
心なしかスター☆の表情にもほころびが。



そしてジンギスカンを食べ終わったと思ったら…



ジャキーン！

マジこの人何枚これ系持ってるんだろw
また、毎回タイミングの良いときに着てくるのがホントにすごいです。持ってますw

やはり変身せずにはいられない。それがスター☆です。



ちなみに背中にはジッパーをアリアリ下ろすとハートブレイクに。…何も聞かないであげてください。

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 | 4月の運勢(4月5日~5月5日) |
|------|-----------------|--|
| 一白水星 | S38/S47/S56/H2 | 本命星と影響と震宮に会しているため、 運気は好調! 流れに乗って 良い出発 が出来る ような月 、 アイデア が採用され 成果 があがるもあろう。ただし 暗剣殺 を受けているために、 軽卒な行動 をとると 逆運 を招く 可能性がある ので注意すべし。 |
| 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 だが 徐々に良くなる であろう。 年盤で五黄殺 を受けているので、 大きな目標・課題 に取り組むときには 注意すべし 。 妨害や邪魔 も入りやすい時期なので、 慎重かつ冷静 に進めることが吉なり。 |
| 三碧木星 | S36/S45/S54/S63 | 今月は中宮に会しているため、 周囲の人から認められ ことも多くなるであろう。 出る杭は打たれやすい 時期なので、 不用意に動かず 一歩引いて物事を 俯瞰的に 見る ことが吉なり 。 口が災い のもとになるので 注意すべし! |
| 四緑木星 | S35/S44/S53/S62 | 今月は年盤の影響で、 楽しみ・実り が多い月なるであろう。ただし 年盤の暗剣殺 を受けているので、 軽はずみな行動 は 禁物 。 万事交替 が激しく 交際範囲 等が 拡大 しますが、 表面的 で 中身が伴わない ことが多いので、 注意すべし! |
| 五黄土星 | S34/S43/S52/S61 | 兌宮に会しているため、 喜び事 が多く 明るい運気 に恵まれるであろう。 目上の引き立て も 厚く 、 昇進 、 昇給 の チャンス をつかむも 可能性 も高いであろう。 経済的・精神的 の両面で 余裕 もてる時期なので、 浪費せず に 日々感謝 を 忘れず 進むべし。 |
| 六白金星 | S33/S42/S51/S60 | 艮宮に会しているため 動かざること山のごとく 、 物事が進まない であろう。ただし 諦めず に 最善 を 尽くす ことで、 大きな変化 も起きるであろう。 ストレス が 溜まりやすい 時期なので、 花見 などをして 心を和らげる ことが 肝要 と心得よ。 |
| 七赤金星 | S32/S41/S50/S59 | 本命星の影響で 拙速 になりがちで、離宮に会しているために 熱しやすく冷めやすい 傾向がでるであろう。 精神的 に ストレス が 溜まりやすい 時期なので、 心身の健康 を 心掛ける ことが 肝要 なり。 発言・成果物 の 再チェック し、 慎重な行動 が吉なり。 |
| 八白土星 | S31/S40/S49/S58 | 坎宮に会しているために 運気は低迷 気味なり、ただし 勉強・スキルアップ を養うには 良い 時期と心得よ。事業・プロジェクト等で八白の特徴である、 「ギャンブル好き」 が できると裏目 に出るので、 八分目 を 目標 に 堅実 に進めることが 吉なり 。 |
| 九紫火星 | S30/S39/S48/S57 | 今月は本命星と相生となり、 運気は徐々に上昇 傾向なり。ただし坤宮に会しているために、 物事の動き が 遅い が、 短気を起こぬ事 が吉なり。 3~5月 は、 本命星と相生 なので、 運気は良い 方向にむいているので、 焦らず着実に 進めることが 肝要 。 |

月盤三碧木星

年盤六白金星

| 南 | | | | 南 | | | |
|----------|--|---|---|---|---|---|--|
| | | 2 | 7 | 9 | | | |
| 東 暗剣殺 | | 1 | 3 | 5 | 東 | | |
| | | 6 | 8 | 4 | | | |
| | | 北 | | | | 北 | |

◆今月のチェックポイント!

今月は「三碧木星」中宮となり、【雷】の卦象の影響により「発見」「ひらめき」が出る月です。
今年「六白金星」中宮のため、「父親」「愛情豊か」という面を強く出ます、「身内のひいきめ」となり・・・
散財せぬよう! さらに「三碧木星」の特徴である「音」「声」が**運勢を強くする**が**形が無いので注意!**
今月は、「**発言**」で得た評価を「**形**」にすることで実績とすべし! 「**有言実行**」を心掛けよ!!

★今月のラッキーテストアイテム

三碧は「ひらめき」「発見」の特徴がある星です。
今月は**探索的テスト**等で、「ひらめき」で想定したテストを超えた不具合を「**発見**」していませんか?
もしくは、**未読のテスト関連の書籍**を通じて新しい「ひらめき」「**発見**」をしてみませんか?

❁お知らせ❁

4月30日(月)に WACATE-Magazine vol.39 発行します！(たぶん)

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、特に戻り確認などを行いません。
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★春眠を楽しむために Tempur 買っちゃいました。暫くは貧乏生活 orz
コヤマ★最近更新が遅くなりがちですみません。次のイベントはどこかなー♪
ぼんぼん★酒の陣で買った麒麟山、気付いたら飲み干されていたという…うわーんwww
あらかみ★「暑さ寒さも彼岸まで」と言われますが、まだまだ寒い日もありますので…
皆さん、お身体ご自愛ください！

★あくづけ★

2012/04/15(sun)はっころ
WACATE-Magazine へんしゅぶ

✉Magazine@wacate.jp
<http://wacate.jp/Magazine/>