ソフトハウス・トランスフォーメーション

2022/7/15（金）
ソフトウェアテストシンポジウム（JaSST）北海道 2022

西　康晴
電気通信大学 大学院情報理工学研究科
情報学専攻 経営・社会情報学プログラム
@YasuharuNishi / Yasuharu.Nishi@uec.ac.jp

Profile

西 康晴
（にし やすはる）

Yasuharu.Nishi@uec.ac.jp
@YasuharuNishi

- Assistant professor:
  - The University of Electro-Communications, Tokyo, Japan
- Vice chair:
  - Software Quality Committee of JUSE (SQiP)
- President:
  - Association of Software Test Engineering, Japan - Nonprofit organization (ASTER)
- President:
  - Japan Software Testing Qualifications Board (JSTQB)
    - 20579 / 36334 (FL) - 717 / 4047 (ALTM) - 548 / 2540 (ALTA) & starting CBT
- National delegate:
  - ISO/IEC JTC1/SC7/WG26 Software testing (ISO/IEC(/IEEE) 29119, 33063, 20246) & SC42/JWG2
- Founder:
  - Japan Symposium on Software Testing (JaSST)
    - 9 (+1?) regions + Review + Online + nano(LTs)
- Founder:
  - Testing Engineers' Forum (TEF: Japanese community on software testing)
- Judgement Panel Chair / member:
  - Test Design Contest Japan, Test Design Competition Malaysia (TDC)
- Vice chair:
  - Society of Embedded Software Skill Acquisition for Managers and Engineers (SESSAME)
- Steering Committee Chair
  - QA4AI Consortium
- Advisor:
  - Software Testing Automation Research group (STAR)

Profile

西 康晴
（にし やすはる）

Yasuharu.Nishi@uec.ac.jp
@YasuharuNishi

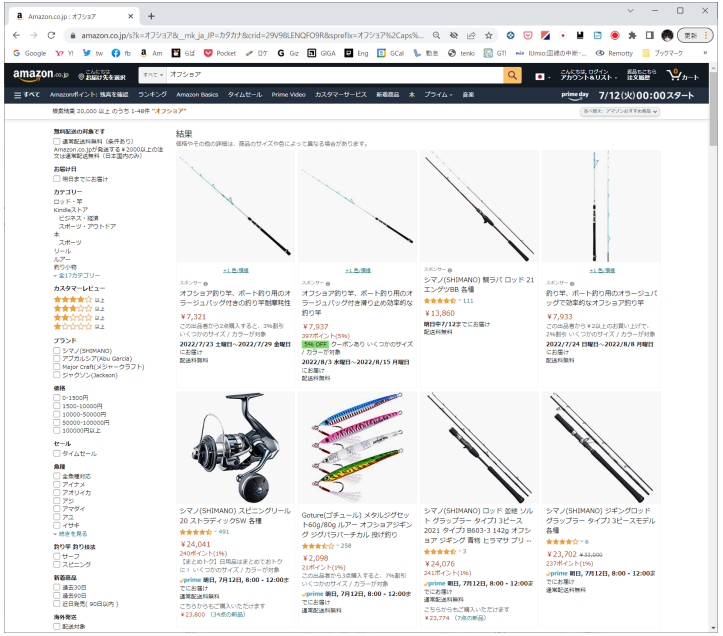
- 本当の本業は「テスト業界ウォッチャー」
- テスト専業のソフトハウス（テストハウス）の
  コンサルティング部長が前職
- 数百名規模のソフトハウスに10年以上も
  事業運営も含めたコンサルティングを提供していた
- 某テストハウスの技術開発に30年近く関与する
- ニアショアを含む様々なテストハウスから、
  テスト組織の経営・事業運営・技術向上について
  年に何度か相談を受ける
- 海外のテストハウスへのコンサルティングも行う

オフショアとは

- オフショアは岸ではなく、船からメタルジグを使って釣るスタイルです
  - 沖に出ることができるため、深水を利用することができます
  - つまり、船から真下にルアーを落とし、ルアーを縦方向に動かしながら魚を狙うスタイルです

（https://www.masakazumaru.net/news/202101/npost-97.php 愛知県の釣り船・雅一丸さんのWebサイトより）

https://www.masakazumaru.net/tyouka.php



オフショアとは

- 岸から沖に向かって吹く陸風のことを指します
  - サーフィン用語です
  - オフショアが吹くと波はなかなか崩れなくなり、面もきれいになるため、適度に弱いオフショアが吹いているのが最高のコンディションとなります（Wikipedia）

http://www.surf-reps.com/news/44947.html

オフショアとは

- 規制が非常に少ない国や地域の金融マーケットです
  - 「国外からの所得」に対して所得税や法人税が安いか、
    まったくかからない「国」や自治権を持った「地域」の金融マーケット
  - タックスヘイブン型といわれるものがその代表的なものです（https://www.ifa-japan.co.jp/what-is-offshore/）

https://rtrp.jp/articles/40730/

ソフトウェア開発における伝統的オフショアとは

- 1990年代後半から、ソフトウェア開発におけるオフショアリングが始まった
  - ソフトウェア開発の一部の工程を海外の会社に任せる形態が主であった
    - 低賃金国
    - 時差が大きな（昼夜が反対の）国
  - 欧米がアイルランドやインドに発注するケースが多かった
  - 経済のふるわない国や発展途上国が
    ソフトウェア産業を振興するための政策でもあった

- 日本では、2000年頃から盛んになった
  - 2022年までずっと続くデフレの真っ只中でコストダウンの要請がとても強かった
    - 時差よりも賃金の要因が大きかった
  - 製造業の「世界最適生産」というコンセプトに影響された企業もあった
    - 「世界中で一番安い部品を買い、世界中の一番安いところで作る」というコンセプト
  - 賃金の上昇に伴ってオフショア先を移転する必要があった
    - インドから中国、ベトナム、ミャンマー、インドネシア、バングラデッシュ…と移転してきている
      - 中国のカントリーリスクを考えて増設・移転した企業も多かった

とても有効な施策だと考えた企業が多かった

その後、廃れる

- 2010年頃から伝統的オフショア開発のデメリットが広く理解されるようになってきた
  - 意外に手間がかかる
    - 日本の下請けだと「これ、よろしく」で済んでいた仕様書を、もっと細かく厳密に書かなくてはいけなくなる
    - 言葉の壁や文化の壁が思ったより大きかった
    - その国の賃金レベルが上がるのが思ったよりも早く、移転が面倒だった
    - スキルが低いので再度日本側でテストをしなくてはならない場合もあった
    - 残業や休日出勤など無理をきいてはくれない
  - そんなに安くならない
    - 仕様書を細かく厳密に書くコストは高単価の日本側タスクになる
    - 言葉や文化の壁を越えるためにブリッジSEが必要になる
    - スキルが低いので高単価の日本側でやり直しタスクが必要になる
    - そもそもオフショア先の選定の際に担当者が接待漬（以下自粛）
  - 発注側の技術が空洞化する
    - そもそもコーディングと設計を完全に分離できる、というのはCOBOL時代の神話に過ぎない
    - コーディングのスキルが低くなると設計のスキルも低下する
    - 色んな事をブリッジSEに丸投げするようになる
  - アジャイル開発や自動化という新しい潮流に対応できない
    - コストよりもスピードの時代になってきた
    - オフショアよりも内製化という流れになった
    - 低スキル作業をオフショアにやらせるよりもコンピュータにやらせる方がDXだよねという時代になった

2022年現在の認識

いまだに伝統的オフショア開発が
有効な施策だと考えている企業は
残念ながら時代遅れも甚だしい

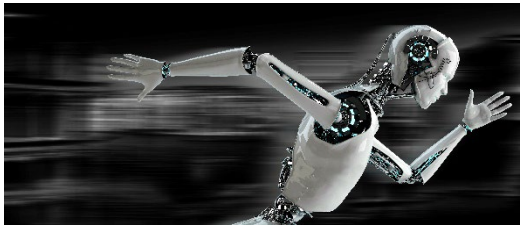
「伝統的」オフショアの後は？

- 「ニアショア開発」を売り込む企業が出てきた
  - 海外ではなく、賃金の安い国内ソフトハウス拠点にソフトウェア開発の一部の工程を任せる形態
  - オフショアの問題点を解決できるという触れ込みである
    - 「これ、よろしく」でいける
    - 言語の壁や文化の壁がなく、ブリッジSEもいらない
    - 超長期デフレという日本の特殊事情のため
      賃金レベルが上がらないので移転の必要がない
    - 残業や休日出勤などの無理をきいてくれる
    - オフショアほど技術力は低くない（？）
  - しかし実際は？
    - 地元の若者の夢や未来と何らかの補助金を引き換えにした愚策である
      - オフショアに仕事を取られた地方ソフトハウスの苦肉の策
      - 手軽に規模を拡大したい格安系ソフトハウスの安直な戦略
    - 「うちの会社は給与は周辺より高い」と自画自賛するソフトハウスもいるが、
      技術力が高まるような施策をきちんと採っていて持続的に給与が上がる考え方のニアショア拠点はほとんどない
    - 発注側の技術力の空洞化は発生するし、アジャイル開発や自動化などにはとても対応しにくい

地方のエンジニアにとって
とても「自分はシアワセだ」と言える
ビジネスモデルではない

「伝統的」オフショアの後は？

- 時代が変化した
  - 日本人エンジニアの質と供給力の低下
    - 世界でも稀に見るスキル獲得に消極的な国民と、シャレにならなくなってきた少子化
  - ビジネスとソフトウェア開発のスピードアップ
    - アジャイル化に伴う内製化
    - 自動化による低スキル作業の消失
  - 技術の高度化
    - アジャイル開発
    - デジタルトランスフォーメーション
    - クラウドコンピューティング
    - モデルベース開発
    - 機械学習
  - リモート環境の充実
    - 東京のエンジニアがその所属のまま地方に住むようになってきた
  - デフレが終わる…？
    - 物価上昇がニアショア拠点の単価向上につながるかは不透明



ニアショア拠点のメリットが
どんどん小さくなっていく

「伝統的」オフショアの後は？

- グローバルソーシング
  - ソフトウェア開発の全ての工程を分割し、それぞれの工程を最適な国で担当するという施策
    - 欧米のグローバル企業の傘下に入った日本企業が必ず直面する
      - テストは大概インドに拠点を作られてしまい、日本からは引き揚げられることが多い
    - ソフトウェア開発そのものが日本で行われる必要がないという議論さえ発生する
      - 日本本社のグローバル企業の場合は、ソフトウェア開発組織に大量の海外人材が常駐するようになったりもする
  - そこまでKKD（勘と経験と度胸）でソフトウェアをつくってきた組織であれば、
    日本からソフトウェア開発が消失してしまうのは自業自得であると残念ながら言わざるを得ない
    - 賃金が安いからではなく、技術力で負けて海外に仕事を取られるという現象は実はこの前から発生していた

- ネオ・オフショア
  - ソフトウェア開発の全ての工程を海外のグループ会社に任せる形態
  - 現地で十二分な賃金を払い、とても優秀な人材を集め、活気があり成長する組織をつくる
  - 夢と未来のある仕事の進め方ができる
    - 工程単位で切り出すのではなく、大きな機能群単位で任せるため、
      顧客のフィードバックを直接もらってどんどん改善していく
      - 日本の開発拠点から指示を受けるのではなく、日本の開発拠点を（友好的）ライバルとして競争する
    - 世界を相手にサービスやプロダクトを開発するビジネスを手がけ、どんどん成長していく

ここまでの現象はニアショア拠点だけでなく
多くの日本のソフトハウスにも実は当てはまる

ソフトハウス・トランスフォーメーション

なんとかしてトランスフォームしないと

日本のソフトハウスは凋落の一途を辿っていく

じゃあどうすればいい？

- 仕事が増えればいい！
  - 流行の技術を追いかけて営業的に仕事をもらおうとするが、ソフトウェア開発そのものが上手になるわけではない
    - そもそもそういうソフトハウスの経営陣や営業は、ソフトウェア開発が上手ということがどういうことなのかが分かっていない
  - アジャイル開発による内製化で（良条件な）案件そのものが減ってきている
    - 悪条件案件に手を出すと、ダメな発注で無理を言われて単価下落圧力をかけられる
- 補助金をもらえばいい！大きな会社の参加に入ればいい！
  - テストハウスがガチャに当たってオカネが入ったときに行う施策ランキング（にし調べ）
    - 1位：何もしない
    - 2位：互換性検証ビジネス
    - 3位：ニアショア拠点の拡大
      - 人員増加や拠点数増加であって、技術力向上ではない点に注意
    - 4位：テストと関係ないビジネスの開始
    - 5位：他社ツールと連携したり、海外ツールのインポーターになったり、自社ツールを開発したりする
  - かくしてガチャは溶ける
    - 成功しないか、もしくは少なくとも既存にニアショア拠点に夢や希望を与えられるオカネの使い方にはならない

何もトランスフォームしなければ、何も変わらない

ソフトハウスの稼働率向上ビジネスをトランスフォームする

- ソフトハウスは基本的に稼働率向上ビジネスであると思われている
  - ソフトハウスは「(技術が進化しないタイプの)装置産業」であると思われている
    - ただし大規模生産設備の代わりに「正社員」が装置になる
      - 社員の給与などは固定費である
      - 需給に合わせて設備(社員)を減らすのが難しい
    - 生産するものの価値を向上するのが難しい
      - 半導体のような技術が進化していくタイプの装置産業の場合は、設備そのものをどんどん更新していく必要がある
      - 技術が進化しないタイプの装置産業の場合は、生産するものの価値を向上するのが難しいので、稼働率をどんどん向上していく必要がある
        - » ソフトハウスの場合、単価を向上するのは非常に難しいという思い込みがある
  - 稼働率を上げようとすると、利益率がどんどん下がっていく
    - 稼働率の向上だけを目指すと、技術力向上に工数を振り向けられないため、新しい案件の単価は上がらず利益率がどんどん下がってくる
      - ニアショア拠点は大都市の仕事を回してもらうので営業力が低い上に、技術が向上しないので新規営業が難しく、単価を下げて仕事を取ってこざるを得ない
    - 既存の案件にしがみつかざるを得ないが、単価は少しずつ下がっていく上に、ずっと低スキルの同じ仕事をするとモチベーションが下がり「こなし仕事」になってしまう
      - 会社が「何かをやろう」と旗を振ってもついてこない社員ばかりになる
      - 低単価の仕事を出し続けるお客様は単価低減圧力をかけてくることが多い
    - 「単価上がらない → 夢がない → 仕事を変えようとしない → 技術が上がらない → 単価上がらない」のサイクルに入り、軽作業事務の会社になってしまう
      - そこにいるのは「他に転職先が無いから」という人たちばかりになる

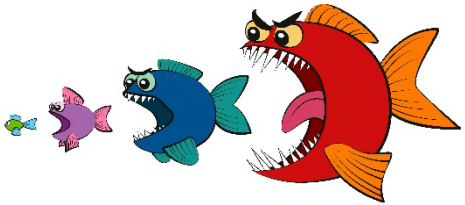
ソフトハウスの稼働率向上ビジネスをトランスフォームする

- 稼働率の向上だけを目指すと、技術力向上に工数を振り向けられない
  - ソフトウェア開発はそもそも生産性に10倍もの開きがあると言われていた技術領域なので、技術力向上や改善の効果が他の技術領域よりも大きい
    - 多くのソフトハウスの経営陣は理解していないので、稼働時間を削って教育や改善に充てることはほとんどない
    - お客様先で技術向上活動をしようとすると「そういうのは自社でやって。オカネ払ってるんだから。」と言われ、自社でやろうとすると「そんなヒマがあるなら仕事して」と言われる
    - 稼働してない時間に自社内で何をしていればいいかの指示はマネジャー層から出ないし、営業からは「稼働『損』」というとても不名誉な呼ばれ方をされる
      - 技術は「お前が仕事を取ってこないからだろ」と思うようになり、営業と技術が離反する
        - » 実際のお客様の困りごとを営業は聞けなくなるし、単価を上げられる（＝お客様の上司が求めている）ことを技術が聞けなくなるので、両者の仕事の質がじわじわ下がっていく
  - 特にこの10年で急に新しい技術が普及してきたため、「技術力を向上すると単価が上がる」のではなく「技術力を向上しないと仕事が取れない」という状況になってきている
    - いつまでもC言語のソースを目でgrepしている時代ではない

ソフトハウスビジネスの種類：稼働率向上型ビジネス

- 自分たちがお客様をどう捉えているかによってビジネスの種類を分けることができる
  - 神様ビジネス
    - お客様の言いなりになって技術的施しを受ける低単価ビジネス
    - 言いなりのことができるまでそこそこ育ててはくれるのでありがたいが、無理を言ってきたり単価低減圧力はかけてくる
    - 育ててもらった結果として低単価ビジネスの規模を増やせるようになるが、規模を増やすと管理コストが跳ね上がり、その分を規模増大で吸収しようとするので、どんどん神様ビジネスから抜け出せなくなる
    - 一見高い技術が必要そうなドメインの業務実績が載っているが、それに対応して必要となる技術についての記述が無い場合はこの可能性が高い

  - グレーターフールビジネス
    - 自分たちよりも低い技術力のお客様と付き合う低単価ビジネス
      - 金融において、高値で株を掴んでもより愚かな者に売れば儲かる、という考え方を「大馬鹿理論」と呼ぶ
    - お客様の技術力がとても低いので、単価は神様ビジネスよりも少し高い
    - 常に自分たちよりも技術力の低いお客様を探す必要がある
      - そういうお客さまは無理を言ってくるし、単価低減圧力を常にかけてくる
        - » お客様の技術力を低いままにする必要があるのでお客様に丸投げさせようとする
          お客様が現場や技術を理解してくださらないので無理を言うようになる



    - その技術がお客様にとって非競争領域ならば単価低減圧力が強くなるにもかかわらず、競争領域になると高い技術を求めるようになって他社に鞍替えされてしまうという実はリスクの高いビジネス

ソフトハウスビジネスの種類：利益率向上型ビジネス

- 自分たちがお客様をどう捉えられるかによってビジネスの種類を分けることができる
  - 相棒ビジネス
    - 共同で技術を開発していく単価上昇型ビジネス
      - お客様と一緒に悩んで技術力を向上しようとする姿勢が信頼を生む
        - » お客様にやり方を教わらない点がポイントである
    - 始めは低単価だが、技術が向上していくとそのお客様内の他の部署に呼ばれたり、技術力を向上する業務に別の名前を付けて単価を上げやすくなる
      - 高信頼性系ドメインでは特に求められる

  - 後輩ビジネス
    - 成長や改善の経験をシェアする比較的高単価ビジネス
      - お客様の現場の問題は結局自社でも発生している、という意外に気付かない事実がある
    - お客様が自分たちの悩みを相談してくるようになり、その解決に向けた基本的な技術方針の策定に関われるので、十中八九その案件は受注できる
      - 景気が悪くなっても切られない受注になる

  - 生徒ビジネス
    - 知識や経験を体系化して名前を付けて売る高単価ビジネス
    - お客様の方を見なくなるリスクがある
      - 自分たちの成長や経験を抜きにして他人がつくった規格などを商材にしようとすると発生する
        - » 必ず取得しなきゃいけないような規格を誰かに作って欲しいなど、受験生ビジネスが欲しいと嘆き始める

  - ファンビジネス
    - お客様が惚れ込んでくださるかなりの高単価ビジネス
    - 外部での講演や執筆、業界全体を引き上げる活動を自社業務として自由に行っている必要がある

どうやってソフトハウスビジネスをトランスフォームしていくか

- 技術力を向上するという姿勢を前面に出す：相棒ビジネス
  - お客様にも自社エンジニアにも分け隔てしない
    - 自分の会社は本気で技術向上をしようとしているんだ、という信頼感を自社エンジニアにもってもらわないといけない
  - 技術力向上のためにきちんと投資をする
    - 徒手空拳で悩むのは技術力向上ではないので、エンジニアが持続的に勉強する仕組みを作り、やる気をだしてもらう
    - 技術力を向上しようとしている社員をきちんと処遇する人事評価制度や組織変更を行うことがベストである
    - 技術力を向上できるのであれば逆に戦略的ディスカウントをするのもアリ
      - ディスカウントした分、技術向上の結果を自社で活用できるような契約にしておく
      - 無理に案件を受注するために値引きをすることとは全く異なるので注意する

- 自分たちが改善し成長する：後輩ビジネス
  - 自組織をどんどん改善して成長させる
    - 改善は、実際に取り組んでみると難しい技術である
      - 改善に手を付ける前にウダウダや議論すらもよい経験になる
  - 改善を業務に埋め込む
    - 改善しなければ仕事が進まない（≠終わらない）ような仕事のルールやプロセスを作る
    - おざなりで改善した雰囲気にしてしまうことは絶対にしないようにする
    - 自分たちが改善する度に楽になったり、ムダが減ったり、スピードが上がったり、といったことを実感できるようにする

どうやってソフトハウスビジネスをトランスフォームしていくか

- 言語化し抽象化し具体化して棚を作る：生徒ビジネス
  - 自分たちがウンウンいいながら頑張って上手くいかせた仕事や、お客様と一緒に開発した技術を書いて/描いてみる
  - 書いた/描いたものに名前をつける
    - 名が体を表すような名前がベストである
      - それを読めばどういう仕組みで何が嬉しいのかが分かる名前がよい
  - （原理的でもいいので）同じ名前がつくようなものは他にないかを探したり、そういうものがないかどうか気をつけながら仕事をし、見つかったらそれを適用して上手くやる
    - 水平展開や横展開と呼ばれる
  - 名前ごとに、その名前と該当する技術や仕事をまとめて整理して、皆に使ってもらえるようにする
    - 棚を置いといてもダメで、定期的に「いいこと考えた会」を開催してトークしたりする

- コミュニティのために汗をかく：ファンビジネス
  - コミュニティを運営したり、良い講演をしたり、ブログを書いたりなど、社外のみんなのために汗をかく
    - 結果としてその人や企業の実力が知れ渡ることになり、「あの人に頼ってみよう」「あの会社ならなんとかしてくれるのではないか」と思ってもらえるようになる
    - 相見積を取られないどころか、言い値で仕事が決まり、成功している限り長く継続発注してもらえる
      - 営業コストがとても低くなり、営業のやり方もプッシュからプル、商材提供からお客様の悩み解決に変わっていく
  - 矛盾ヲ抱擁スル
    - 名前を付けた良い技術であっても、異なる技術を同じ現場に適用する際には矛盾が起こることがあるので、そうしたところを解決できるようにしておく
      - アジャイル開発とモデルベース開発、高度な品質保証の組み合わせなど

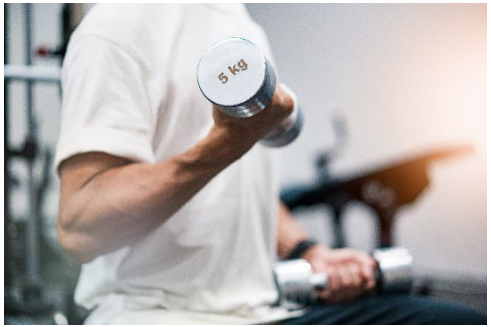
ソフトハウス・トランスフォーメーションに必要なマインドセットは？

- 学び続け改善し続ける組織をつくる
  - 労働ではなく知恵を売るためには、皆で知恵を仕入れて皆で知恵を生み出し続ける組織と風土が必要である
    - 知恵を仕入れて知恵を生み出すことを評価する人事評価制度と組織構造をつくることで経営から価値観を伝える
    - 皆で知恵を生み出し続けるために、一人一人が技術的好奇心を持ってたくさん話す風土や習慣を醸成する必要がある
- お客様の本当の悩みに向き合う
  - お客様の指示に従うのではなく、お客様の本当の悩みを感じ取り、向き合い、共感できるようにする
    - 技術と営業が良いタッグを組まないとお客様の本当の悩みは分からない
- シェアを取ろうとせずパイを大きくする
  - 目の前や見込みのお客様との案件ばかり考えるのではなく、
    産業やコミュニティ全体に貢献することで自然に案件が多くなる方法を考える
    - パイが大きくなった時に自然に自社に仕事が来るように技術を高めておく
- 「スマートな愚直」を心がける
  - 脳みそから血が出るほど考えて考えて考え抜いて最適なアプローチを決めるとともに、
    どんどん発生する困難に負けずに現場で肩まで泥に浸かって諦めず絶対に歩みを止めない
    - “ローマは1日にしてならず”
- 自組織のアイデンティティやDNAを常に自らに問いかけ続ける
  - 「私たちはどんな会社？」「私たちは何に価値を置くの？」「私たちにとって捨てるべきものは何？」など
    - 稼働率向上型ビジネスをしているソフトハウスのほとんどは、アイデンティティやDNAが「無い」
  - アイデンティティやDNAを端的に表して言葉が皆の「口ぐせ」になるくらい浸透させる
    - 組織文化の変革に必要なのはそういう泥臭いことの積み重ねであって、格好いいけど空虚な経営メッセージではない

2022年現在でトランスフォーメーションに有望なテスト・QAの技術

- ツールに依存しないCI/CDパイプラインの構築とツール選定
- アジャイル開発におけるきちんとした品質保証
- ふり返り・改善の方法の体系化と実践的技術の構築
- パイプラインのクラウド化と並列協調化
- テスト開発に沿った縦の自動化パイプライン（MBT→KDT→CI/CD）の構築
- （単なるテスト観点一覧ではない）テスト観点のモデリング技術の確立
- バグ分析からのアンチパターン化とシフトレフト
- モデルベース開発における品質保証
- 製造業向けのテストアーキテクチャ設計と自動化パイプラインの構築
- 高い品質/信頼性/安全性などの達成と認証取得のロードマッピング
- 迅速かつ的確な出荷判断のための（品質）ダッシュボードの構築
- サービス価値や事業パフォーマンスまで保証するQAの確立
- シフトライト・カオスエンジニアリング技術の獲得
- フレイキーなテストを扱う技術の獲得
- 機械学習による自動化テストの改善
- 機械学習ソフトウェア・サービスの品質保証の確立

などなどたくさんあります




覚悟を、決めましょう