



時代遅れでブレーキ扱いのQAから、 イマドキでアクセルとなるQAへの脱皮



2023/7/24 (月)

SQiPソフトウェア品質保証部長の会 第14期成果発表会

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
西 康晴

@YasuharuNishi / Yasuharu.Nishi@uec.ac.jp

自己紹介

- 大学の研究者:
 - 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
 - 電通大は先日まで(今も多分)品質管理の教員が日本で最も多い大学である
- 翻訳者・監訳者:
 - 実践ソフトウェアエンジニアリング(オーム社)、セーフウェア(翔泳社)、ソフトウェアテスト293の鉄則(日経BP)、基本から学ぶソフトウェアテスト(日経BP)など
- 理事長:
 - NPO法人 ソフトウェアテスト技術振興協会 (ASTER)
- 運営委員長:
 - 日本ソフトウェアテスト資格認定制度 (JSTQB)
- 国際規格エキスパート・国内小委員会主査:
 - ISO/IEC JTC1/SC7/WG26 ソフトウェアテスト (ISO/IEC/(IEEE) 29119, 33063, 20246)
 - ISO/IEC JTC1/SC42-SC7/JWG2 AIシステムのテスト (ISO/IEC AWI TS29119-11)
- 設立メンバ:
 - ソフトウェアテストシンポジウム (JaSST)
- 主宰:
 - ソフトウェアテスト技術者交流会 (TEF)
- 審査委員長・審査委員:
 - テスト設計コンテスト(日本)・Test Design Competition Malaysia
- 運営副委員長:
 - 日本科学技術連盟 ソフトウェア品質委員会 (SQiP)
- 副理事長:
 - NPO法人 組込みソフトウェア管理者技術者育成研究会 (SESSAME)
- 世話人・お世話係:
 - AIプロダクト品質保証コンソーシアム (QA4AI) / JaSST nano / アジャイルSQC研究会
- アドバイザ:
 - ソフトウェアテスト自動化研究会 (STAR)



品質マネジメントは持続的活動である

- 品質マネジメントの定義(SQuBOK 1.2/品質のマネジメント)

- 組織を長期的・安定的に存続させるには、組織の活動の主たるアウトプットである製品・サービスを顧客に提供し、それによって対価を得て、そこから得られる利益を再投資して価値提供の再生産サイクルを維持することが必須である。
- そのためには、組織が提供する製品・サービスが長期的に幅広い顧客に満足を与え続けなければならない。これを実現するための武器が「品質のマネジメント」である。

- 持続的活動に必要なもの

- 伝統と進化
 - 寿命の長い企業は、何らかの伝統を持ち、絶えず進化を続けている
 - 進化を止めたものは悪い意味で「伝統芸能」であり、限られた世界でしか市場での競争に勝ち抜けず、国家の庇護を必要とする
- スパイラル的進化
 - 事業環境や技術が変わっても、企業は世代ごと(数年から十数年おき)に本質的に同じ問題を解き続けている
 - したがって逆タイムマシン経営もまた重要である
 - 「われわれが歴史から学ぶべきは、いかに人々が歴史から学ばないか、ということである」(ウォーレン・バフェット)
 - ある世代での進化を怠ると、次の世代での進化に追いつけない
 - 周回遅れは周回遅れであって、停まっていたからトップに立てるわけではない



品質マネジメントの伝統: 日本における品質管理(TQM)の歴史

- 戦前から統計的品質管理(SQC)は日本で適用されていた
 - 品質管理の始まり: ベル研究所のShewhartが管理図を考案(1924)
 - インターンに来ていたのが学生だったDemingである
 - 東京電気(現東芝)の石田保士が電球の製造に統計的方法を導入(1929)
 - Shewhartとは別に管理図相当を考案し統計的品質管理をしていた(1931)
- GHQとデミングと神8
 - 戦後すぐGHQのCCS(民間通信局)が電話回線の故障を低減するために企業に講義をしていた
 - ソフトウェアのQCが盛んだったのが電電ファミリーだったのは偶然ではない
 - 日本の国勢調査の計画立案および支援のためにDemingが来日し、統計的品質管理の講義を行った(1950)
 - 産官学からデミングの講義を受けたQCRGの8名が日本における品質管理の研究・実践・普及を始めた
 - 石川馨(東大)、朝香鐵一(東大)、水野滋(東工大)、木暮正夫(東工大)、後藤正夫(官僚)、東秀彦(国立工業技術院)、三浦新(三井化学)、渡辺英造(三菱金属)
 - 彼らはQCだけでなく、統計学、OR、IE、RE(信頼性工学)など様々な分野で活躍した
 - デミング自身はその後もずっとアメリカでは無名の研究者/コンサルタントだったらしい
 - 動画も残っている:「新しい品質管理」(1960年代初頭)
 - <http://www.kagakueizo.org/movie/industrial/5181/>
- 品質管理は日本の製造業企業にどんどん広まっていく
 - 日本的品質管理は一朝一夕に完成されたものではなく、偉大な先人たちの叡智の集積である

これらを知っている
ソフトウェアQAは
どれくらいいるでしょう?



<https://www.juse.or.jp/deming/award/>



品質マネジメントの伝統: 日本における品質管理(TQM)の歴史

- そして日本はSQCを全社的品質管理(TQC)に **模倣改造** 発展させ、品質立国となる

- アメリカのTVドキュメンタリーで放映されるまでに至った

- “If Japan can ..., why can't we?” (1980)
 - このドキュメンタリーでデミングは全米で引っ張りだこの有名なコンサルタントになっただけでなく、
 - 実際には、日本の工場や企業ではQC、IE/TPS、OR、REなどが渾然一体となって進化した
 - 何をやらうとも、理想の現場をイメージできていなければ上手くいかない

- 全社的品質管理(TQC)

- 統計的品質管理の様々な考え方を企業全体に適用しようという考え方
 - この考え方そのものはFeigenbaumが1961年の本で紹介していたが、当時イケイケだったアメリカの産業界にはあまりウケなかったらしい
 - NBCのドキュメンタリーが示すように、1970年代後半からアメリカの産業界が傾き、「日本に学べ」ブームが起きたらしい

- デミングの功績を称えた「デミング賞」は日本の品質管理を牽引する一つになった

- 日本には優れたTQCを行った企業に与える「賞」はあるが、認証制度は存在しない
 - 品質管理は各企業で咀嚼し発展させるべきもので、これをやれば十分というものではない
 - TQCを逆輸入したアメリカはアメリカ版TQCであるシックスシグマやアメリカ版デミング賞であるマルコムポルドリッジ国家品質賞を生み出す
 - デミング賞は経団連会長が授与するが、MB賞は大統領が授与する
 - MB賞は製造業だけでなく様々な産業に与えられた

- 総合的品質管理(TQM)

- TQCを体系化し、経営の品質に拡張し、製造業以外の様々な企業に応用したものがTQMである
 - ソフトウェア、建設、医療、サービスなど

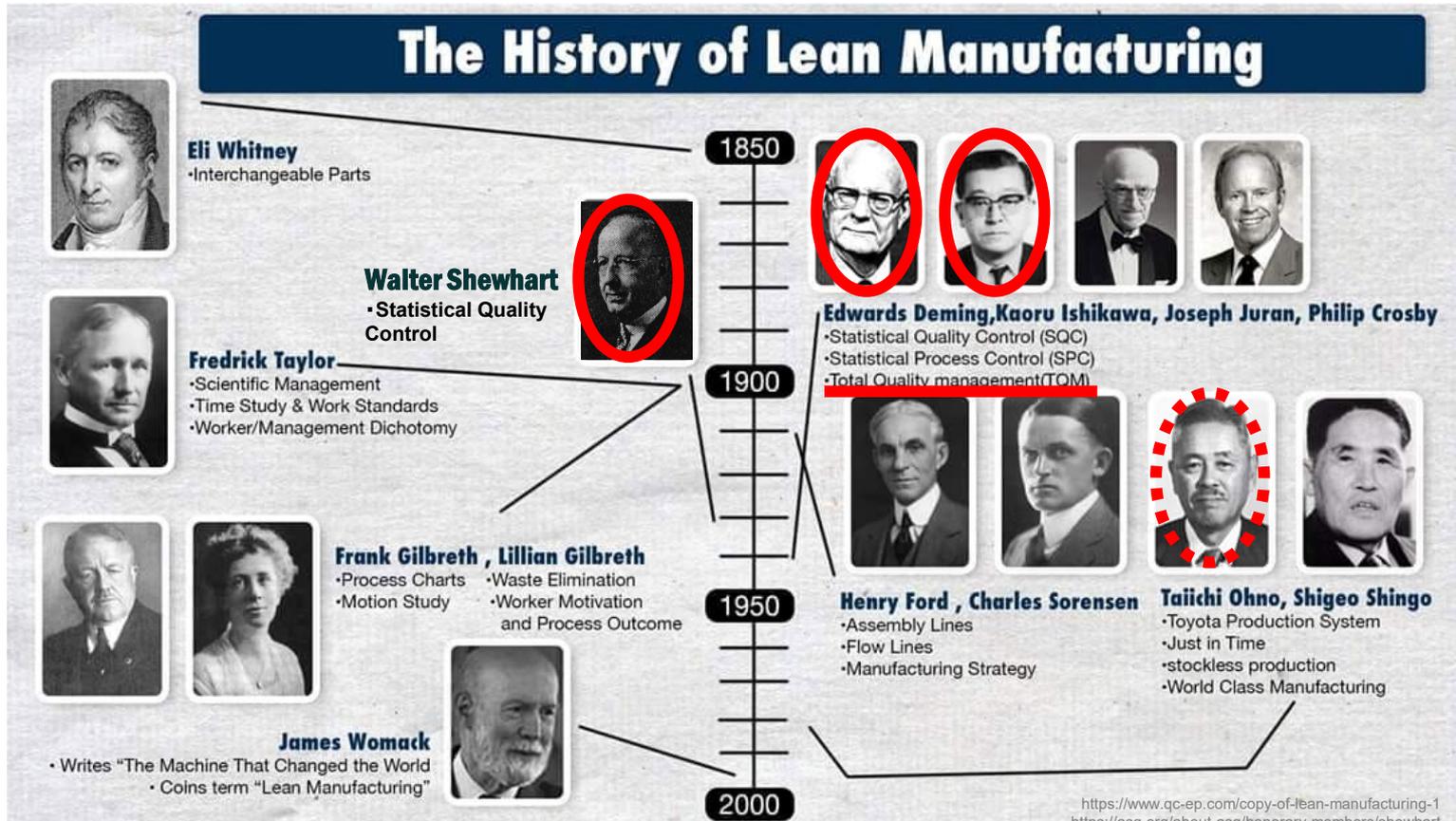
これらを知っている
ソフトウェアQAは
どれくらいいるでしょう？



<https://twitter.com/MarkGraban/status/6680668019516522496>



品質マネジメントの伝統：TQMは確かに海外でも認知されている



品質マネジメントの伝統: 日本的品質管理(TQM)の原則

※諸説あり

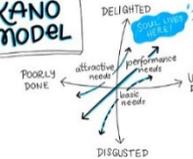
- ・ 品質
- ・ 品質第一
- ・ カイゼンサイクルと標準化
- ・ 方針管理
- ・ 目的指向とプロセス主義
- ・ 水平展開による品質の作り込み
- ・ 全員参加と「後工程はお客様」
- ・ 事実に基づく管理と5ゲン主義
- ・ 自律と小集団活動
- ・ 人間性尊重
- ・ 悪さの知識

これらを知っている
ソフトウェアQAは
どれくらいいるでしょう?



https://www.juse.or.jp/qc_circle/case/

The
KANO
MODEL



<https://www.agilityiq.io/product-management/the-kano-model-for-prioritization/>



ソフトウェアQAは
品質マネジメントの伝統を
ほとんど知らない



では、ソフトウェアQAはTQMを学べばいいのか？

- 残念ながら、TQMをソフトウェアQAが学んでも、それだけではほぼ役に立たない
 - (長い歴史を持つため)ソフトウェア開発の方法論ほど体系化されていないし、スッキリ整理もされていないし、「これをやればTQMをきちんとやったことになる」ものも存在しない
 - もちろん教科書やセミナーはある
 - 国際規格ISO9000sはTQMではなく、単なる入口にすぎない
 - 整理されていないからこそ、よく考えて自分ごととして考えるようになる、というパラドックスがある
 - 誰かが考えたTQMを導入するというマインドではなく、自分たちなりのTQMを構築するというマインドが必要である
 - 組立加工機械の事例が多くソフトウェアの事例がとても少ないので、読み替えにくい
 - そもそも専門用語が分からない
 - 同じ用語(例:プロセス)でもソフトウェア開発とは意味が異なる
 - 工場での製造・生産が主な例のため、(機械でいうところの)企画や設計が主なソフトウェア開発にはそぐわない
 - 統計に頼り過ぎている
 - ソフトウェア開発のメトリクスは人間系、社会系や認知系の要因が多いため分布がよく分からず誤差が多い
 - 自然言語でのデータが多いソフトウェア開発には伝統的統計学は適用しにくい
 - 自動化されたクラウド上の開発のように、膨大なデータが入手できるのなら統計は強力な武器となるが、残念ながらTQMで教えられる統計と同じかどうかよく分からない
 - 工場での製造・生産とソフトウェア開発の本質的違いを解説している資料は皆無である
 - 工場での製造・生産の作業のように手順を決められることと、考えるという手順を決めきれないことは本質的に異なる
 - 反射的に間違えることと、よくよく考えて間違えることは本質的に異なる
 - したがって、バラツキの低減でも標準の遵守でもない、異なる考え方が必要になる
 - セパレーションしやすいので複雑度が管理できるものと、セパレーションしにくいので複雑度が極めて高いものはずいぶん様相が異なる



それでも、ソフトウェアQAの先人たちは学び、実践し、苦闘してきた



https://www.i-juse.co.jp/statistics/member/essay-ayatomo_kanno.html

「次工程はお客様
って何ですか？」

「CMMって
何ですか？」

- 黎明の時代:SPC時代 – 汎用機ベンダと神々の宴
 - 1980年にSPC (Software Production Control) 活動が日科技連で始まった
 - 菅野文友先生の強力なリーダーシップと旧汎用機ベンダ・電電ファミリが主体で議論を進めていた
 - 業界全体が神々の宴だった: 岸田孝一、松原友夫、菅野文友...
 - ソフトウェア工学とTQC/TQMの結婚
 - 汎用機のOSやミドルウェアの開発は、TQC/TQMが適用しやすかった
 - わりと成功を収めた
 - クスマノのレポート: 日本のソフトウェア品質は他エリアよりも一ケタ高い
- 苦難の時代:SQiP時代 – タコプロとブラック労働
 - 1990年代中盤から技術や産業構造、雇用形態が変化してきた
 - ネオダマ(ネットワーク、オープン、ダウンサイジング、マルチベンダ/マルチメディア)と言われていた
 - しかし本質的なのはむしろタコプロ(多重下請・工数精算・プロジェクト制)になったことであった
 - タコプロ a.k.a. ウォーターフォール in アジャイル
 - TQC/TQMをほとんど知らない学ばない腹落ちしていないソフトウェアQAが品質技術を衰退させていった
 - そもそもタコプロはソフトウェア開発を成功させにくい産業構造だった
 - タコプロに抗うように日本版CMMやISO9000sで重量化・形骸化・無目的化・無知見化が進んでいった
- 再構築の時代:ADAM時代 – 品質管理の消失から再構築へ
 - 2010年前後から本格的にADAM(アジャイル、DX、AI for SE、機械学習プロダクト開発)が普及してきた
 - そこにはQAという単語は存在せず、SQiP活動もほとんど存在感を示せなかった
 - 2000年代までの日本の知見を継承するQAがほとんどいなくなった
 - 2020年前後からアジャイル開発の失敗例が世に見られるようになり、品質管理の再構築が話題となってきた



そしてソフトウェアQAの苦闘は続いている

- 再構築の時代 – ソフトウェア品質管理の消失:2010年前後
 - アジャイル開発のプラクティスに、品質管理の mindset やプラクティスが結構入っていた
 - レトロスペクティブ、改善、ペアプロという名の持続的レビュー、TDD...
 - つよつよエンジニアばかり露出する時代なので「ちゃんと作ればテストやQAはいらない」と言われていた
 - 初期のスクラムガイドにテストやQAというロールはきちんと明記されていない
 - アジャイル開発のQAにしっかり取り組もうとするQA専門家はごく少数だった
 - ウォーターアジャイルフォール開発にタコプロQAを適用して仕事をした気になるQAが多かったように思う
 - インチキなコンサルタントに振り回される業界もあった
 - 「QMでは安全は保証できません！」「そこはQMでしっかりやってもらわないと...我々は分かりませんので...」
 - 「QA=テスト」という世界の常識(?)に追従するエンジニアが多かった
 - 昔から洋物好きな業界なのでね...
- 再構築の時代 – ソフトウェア品質管理の再構築:2020年前後
 - アジャイル開発が理想通りいかないという例が世間で見られるようになってきた
 - 1986年に提示された「銀の弾丸はない」という定理は、今でも有効だった
 - 自動化やパイプライン、テストピラミッドなどテストやQAの技術が高度化し、ミッションクリティカルや大規模なプロダクトにも適用されるようになってきた
 - シフトレフトやホールチームアプローチなどが海外で喧伝されるようになってきた
 - 海外でもてはやされている専門家はQuality Engineeringなどと呼んで講演しているが、残念ながらシフトレフトやホールチームアプローチをふんわりとしか理解していない、という例もある
 - タコプロなQAにとって破壊的な技術進化が訪れた
 - AIプロダクトのQAはどうすればいいのか？
 - QAのDX、DXのQAはどうすればいいのか？



しかし再構築は(日本では)ほとんど進んでいない

- 品質保証の伝統を軽んじる一方で、(ソフトウェア)品質保証の進化を怠っている企業や組織が非常に多い
 - 事業環境や技術が時代によって変わっても、今までやってきたことに拘泥してしまう
- なぜ進化させようとししないのか？
 - 品質保証とは、「守る」「守らせる」仕事だと誤解している人が極めて多いから
 - その人たちは、何かしらの「正解」「模範解答」があると思っているくらいがある
 - 品質保証は「後の先」の技術なので、矜持が足りないと「後の後」になってしまい保守的でブレーキになる
 - 伝統を学ばず実践しようとしないので、いつまで経っても本質が分からない
 - バズワードに踊らされ現場に突き上げられ「同時代性の罠」にハマる
 - 洋モノという蟹気楼を無批判に追いかけ、インチキなコンサルにハマる
 - 指摘しても「だってしょうがないじゃないか」と口を尖らせる
 - 事業環境や技術が変化しているのに進化しようとしないと、内向きで引きこもりになる
 - いつまで経っても全社品質保証部とソフトウェアQA機能が融合せず乖離していたりする
 - 品質保証部長の仕事は経営者に説明することだ、といった内向きで引きこもりな思考は実際に耳にする
- その結果...
 - 時代遅れでブレーキ扱いのQAになる
 - 社内で(ソフトウェア)品質保証の位置づけがとても軽くなる
 - 水面下で組織の弱体化が起き、品質トラブルが発生し、経営的な問題になってしまう



自分たちでソフトウェアQAの(再)構築をしよう

- 外国人に「Whole Team Approachが大事」とか言われて感心してるのは恥ずかしいと思ひましょう
 - 「品質を上げようとするスピードが上がる」と言われて「そんなの当たり前でしょ」と返せないのも恥だと思ひましょう
- ソフトウェアQAの(再)構築をするためには、伝統的TQMをきちんと腹落ちしないといけなひ
 - 品質保証部長やCQOのマインドを変える
 - 内向きで引きこもりなソフトウェアQAから、伝統を腹落ちし進化し続けるソフトウェアQAになるというマインドにシフトする
 - 既存の旗を振っておけば楽ができる、というマインドでは話にならない
 - TQMを勉強し、実践する
 - 詳しくは日科技連さんに聞いてください...
 - そのまま話を聞いているだけでは、ほとんど役に立ちません
 - 自分たちの組織能力が上がるように実践しましょう
 - 部長の会やSQIP研究会で、もしくは新たな研究部会を立ち上げて、社外の人たちとソフトウェアQAのためのTQMの勉強・実践の会を推進する
 - きちんとした専門家を呼ばないと、ちんぷんかんぷんになってしまうだろう
 - 大学の先生によるソフトウェア品質保証の指導会を行い、自社のTQMの活動の壁打ちを行う
- そして、伝統そのものを進化させていく必要がある
 - 伝統芸能化したTQMそのものを進化させる必要もある
 - TQMのコミュニティが内向きで引きこもり、祖父の遺産の焼き直しばかりになってしまっているのて、スタートアップやアジャイル、ソフトウェアやサービスといったイマドキの事業環境や技術が伝統芸能の進化を促さなくてはならないだろう



大学の先生による品質保証の指導会

- 見識のある専門家に毎月「壁打ち」をするようなイメージ
 - QAの副業でこのタイプもある模様
 - 短いお付き合いでは残念ながら成果は期待できない
 - 指示したことしかできない派遣テスターを数人雇うより長期的効果は高い
- もちろん品質技術の共同研究をしてもよいし、現場のカイゼンをコンサルしてもらってもよい
 - ただし企業側の現場にカイゼンしたいという強い思いが無いと絶対に上手くいかない
 - やる気のある若い人に業務で品質技術での博士号や修士号を取らせる、という手もある
- 若い先生にロングスパンで指導会をお願いすることで大学の先生を育てている企業もある
 - 業界全体の品質技術のリードや向上、社内での品質保証部のプレゼンス向上など、大学の先生がきちんと産学のことを理解してよい仕事をするとう嬉しいことは結構ある
- 多くの企業は、大学とどう付き合いばよいか分からない
 - まるで案件のように「共同研究」を発注し、イマイちな結果が出てくるが、文句も言えない
 - 発注契約やNDAなど、実は大学側で様々な契約形態やフォーマットを持っている
 - 「業者」ではないので、企業側に熱意がととも必要である
- ソフトウェアの世界には、多分、ほとんど存在しないプラクティスである
 - DXでAIの時代になって、欧米のソフトウェア工学は産学連携がととも進み、企業は大学の最先端の技術をどんどん使うようになってきている



ソフトウェア品質保証の進化

- 進化には研究開発が必要である
 - 皆さんの品質保証部では、研究開発機能を明示的にアサインしていますか？
 - 要素技術や開発技術の研究開発ではなく、品質技術の研究開発である
 - 他社の品質保証を担うはずの第三者検証企業で研究開発機能を持つ企業はほとんどない
 - デミング賞受賞企業によって生み出された品質技術はいくつもある
 - 勉強会に参加するのは当たり前の行動であり、研究開発機能ではない
 - 片手間や残業時間、思いつきではダメ
 - 研究開発しないQAは「時代遅れでブレーキ」になる
 - 一部の研究会や部長の会の会グループのような活動のように、ただ集まっておしゃべりした内容を紙に書いただけのものは研究開発ではない
 - 研究開発する、というのは、研究開発に予算を割り当てる、という意味である
- 進化には「あり方」や「戦略」を考える人たちが必要になる
 - 皆さんの企業における品質保証の「あり方」って何ですか？
 - ただテストの作業を効率的に行う組織ですか？
 - シフトレフトやフロントローディングが根付いた組織ですか？
 - ただ他の企業を真似ているだけではありませんか？その企業特有の「ストーリー」はありますか？
- 開発技術が新しい技術を取り入れたり適用するよりも先に新しい技術に対する品質技術を確立し、新しい技術による製品開発を後押ししてやる必要がある
 - 迷える開発に対する「安全装置付きアクセル」になる必要がある



アクセラとなるQAが持つべき「先行構想能力」

- 「後の先」とは
 - 相撲の立ち合いで
「相手より一瞬遅れて立つように見えながら先手を取る極意(双葉山のWikipedia)」
 - 本来は剣術などにおける返し技(ボクシングにおけるカウンター)の意味らしい
 - 相手をよく見ること、遅れて立ったのに先手を取れる俊敏さ、それらを支える日頃の鍛錬が必要
- 特にDXな時代では、QAは開発に対して「後の先」で臨むのが理想である
 - 「後の先」: 開発が新しい技術や開発手法や価値などを採り入れたら、すぐにそのQAの技術やプロセスを確立する
 - 開発がいまどういう状態で、何を目指そうとしていて、何ができていて何ができていないのか、をよく見る
 - 開発が採り入れた技術や開発手法や価値について、俊敏に把握してQAの技術やプロセスを確立する
 - 開発が何を指すべきか、何を採り入れるべきか、その時に何が問題になりそうか、を日頃からアンテナを立てて情報を収集し、QAとしての技術やプロセスを研究開発しておく
 - 技術がスパイラルであるならば、次に何が重要になるかは推測しやすい
 - QAの技術のスパイラルはさらに、開発技術のスパイラルの後を追うことになる
 - 「先の先」(開発に「こんな技術や価値があるよ」と提案するQA)はもちろん悪くないが、あまり現実的ではないかも...
 - もし日本人がオリジナルが苦手と真似が得意というなら、「後の先」戦略は得意なはずである
- しかし実際の品質保証組織は、開発が新しいことを取り入れたのに...
 - 「後の後」: QAが追いつかないから責任あるリリースができない
 - 「後の不」: QAはそれを理解できず適用できず開発に丸投げする
 - 「後の無」: QAは今まで通りでいいと知らぬ顔を決め込み硬直化しサイロ化する



矛盾ヲ抱擁スル

- 矛盾すると思われるものを両方とも飼い慣らすことが
伝統(右手)と進化(左利き)を両立させたい
質の高い組織には必要になる
 - 外モジュラー・中インテグラル
 - 5ゲン主義 = 3ゲン(現場現物現実) + 2ゲン(原理原則)
 - サーバント・リーダーシップ
 - 大局観を描くイマジネーションと、緻密なロジック
 - 分散型俯瞰
 - あらゆる先入観を取り除きあらゆるものに興味を持つ自由さと、徹底的なこだわり
 - しなやかに変化できる能力と、困難から逃げない姿勢
 - スマートな、愚直・泥臭さ
 - 事実を基にしたロジカルな予測による予防と、周到に準備されたレジリエンス
 - データやエビデンスによる論証・説明責任と、納得感を共感できるストーリー
 - 仕事がうまく回っているという安心感の追求と、明日にも退場させられるのではないかという危機感の維持



強い製造業企業は両利きを実現してきた

- 経営学者や経営コンサルタントは、右利きと左利きは相反すると主張することが多い
 - 多くのイノベーション論では、カイゼンする組織はイノベーションできないと一刀両断される
 - 確かに、右利きを続けるとある種の限界のようなものに達してしまい、カイゼンしたふりや捏造、組織的な硬直や官僚化が発生する場合も多く見られる
 - しかし右利きだけでは競合他社に同水準の製品をすぐに開発されレッドオーシャンになってしまう
- 実は日本の強い製造業企業(の強い現場)は、両利きを実現することで勝利してきた
 - 枚挙を上げれば暇が無い
 - トヨタ生産方式はどのようにして生まれたのか
 - 日本の自動車系企業はこれまでいくつの「世界初」「世界一」を生み出してきたか
 - そもそも「漫然と目の前の業務に習熟する」ことは右利きではないし、右利きから左利きを生み出すには経営陣・上級管理層・現場の全てに強い動機や危機意識が必要である
 - コマツの「ダントツ」は米キャタピラーとの(極めて分が悪いと誰もが思っていた)戦いで生まれた
- ただし、ルールや技術の変化の頻度(や程度)がケタ違いに高く大きくなっているため、より強い動機や危機意識、決断とスピードが必要になる
 - 例えば車載電装品は車載部品の中でも電気→電子→半導体→ソフトウェアとかなり変化してきた部類だが、その変化の程度とスピードはCASE・MaaSの到来と比べものにならない
 - 国内にもデジタルツインが実現できている製造業がある
 - 左利きを身につけやすい「筋のよい」右利きの技術を身につけられる体質であることが重要になる



「ちゃんと」してない組織の社内で飛び交う危険な言葉

- 「ちゃんと」している組織(強い組織)で交わされると両利き力が上がっていくが、「ちゃんと」してない組織(弱い組織)の社内で発せられると組織が腐っていく言葉がある
 - 「それを何とかするのがお前の仕事だ」
 - 強い組織: 現状の常識から離れて新しいアプローチを探索しろ
 - 弱い組織: 無理しろ
 - 「お前の責任だ」
 - 強い組織: (部下に)お前に権限を与えるから周りを気にせず思いっきりやってみろ
 - 弱い組織: (上司である)自分に責任はない
 - 「(無理と思える目標に)チャレンジしろ」
 - 強い組織: 現状の仕事の進め方に悪い意味で満足してしまっているから、負荷をかけて惰性になっているところを見直そう
 - 弱い組織: 良い結果を捏造しろ
 - 「お前はと思う」
 - 強い組織: 担当だからこそ分かることがあるだろうし、温めているアイデアがあるだろうから、オレのアイデアと比べてみよう
 - 弱い組織: オレは考えたくないからお前が考えろ、それがボトムアップだ
 - 「なぜだ」
 - 強い組織: 真の原因を探ってパターン化し対策を講じて、もうこのことで誰も困らないようにしよう
 - 弱い組織: お前の責任だオレは悪くない謝れ
 - 「オレに分かるように説明しろ」
 - 強い組織: よりよいシンプルなロジックにして皆に納得してもらいやすくなるよう、お前のアイデアと一緒にカイゼンしよう
 - 弱い組織: オレの顔を立てろオレを通さずに仕事をするなお前はオレの道具だ



<https://ja.wikipedia.org/wiki/隠し絵>



我々が進化させるべきソフトウェア品質技術の例

- 「品質保証」という概念の整理と、あり方の類型の提示、戦略の立案、文化の醸成
- プロダクトレベルから事業レベル、経営レベルにトランスフォームする品質保証の形
 - ソフトウェアQAが経営の質、経営の問題点を経営者と一緒に追いつける
- (ミッションクリティカルな)アジャイル開発における品質保証
- 完全自動化を目指すデジタルツインなモデルベースQAパイプライン
- かけ声だけでないWhole Team Approach
- イマドキのソフトウェア開発における悪さの知識のパターン化
 - ダメなアジャイル、ダメなDevOps、ダメなサービス、ダメなスクラム、ダメな設計...
- 価値と品質、事業やサービスと開発、開発やQAのメトリクスなど包括的なメトリクススイート
- AIにおける品質保証の技法と具体的事例
- 保証する、というのは一体どういうことなのか、の再考
- 思考の質の向上



私の周りでの研究開発の例

- 「品質保証」という概念の整理と、あり方の類型の提示、戦略の立案、文化の醸成
 - 例) QMファンネル・QAスタイルファインダー(SEA-SIGSQA)、技術ロジスティックス
- プロダクトレベルから事業レベル、経営レベルにトランスフォームする品質保証の形
 - 例) DX4TQM
- (ミッションクリティカルな)アジャイル開発における品質保証
 - 例) 乞うご期待(某社との共同プロジェクト)
- 完全自動化を目指すデジタルツインなモデルベースQAパイプライン
 - 例) QualiTrax
- かけ声だけでないWhole Team Approach
 - 例) 納得感の共感(SEA-SIGSQA)
- イマドキのソフトウェア開発における悪さの知識のパターン化
 - 例) ソフトウェアトラップ / 不具合モード
- 価値と品質、事業やサービスと開発、開発やQAのメトリクスなど包括的なメトリクススイート
 - 例) 乞うご期待(SQiPアジャイルSQC研究部会)
- AIにおける品質保証の技法と具体的事例
 - 例) メタモルフィックテストング、Boosted Exploratory Testing
- 保証する、というのは一体どういうことなのか、の再考
 - 例) 保証性
- 思考の質の向上
 - 例) ロジックキューブ



SQiPや部長の会に期待すること

- ソフトウェア品質保証に関する活動の集積と進化のための業界横断的なメタコミュニティをつくろう
 - 現状では、ソフトウェアの品質に関する活動が分散してしまって、どこの何をみればいいのか分からない
 - SQiP系、ASTER系、STAR、JASPIC、SEA-SIGSQA、JSSST/MLSE、アジャイル方面の各種コミュニティ、Googleに学ぶ人たち、機能安全、ゲーム方面...
 - SQUBOKは本当に読まれ活用されているのか？単なる資格試験のためのシラバスになっていないか？
- 各自の組織で新しい技術を生み出し、新しい技術を適用し、事例を公開しよう
 - 基本的に品質技術は生産技術のように外部から見えない
 - 製品をリバースしても、どんなテストをしたかは本当のところ分からない
 - だから業界内で、業界横断的に、積極的に技術を共有していかないといけない
 - そもそも、自分から情報発信する人のところには、色んな情報が集まるものである
 - 例)テスト会社の上手な使い方って、知ってます？
 - QAの部長やメンバをコミュニティ活動に業務として参加させるのはとても意味がある
- 我が国のソフトウェア品質保証における伝統と進化を持続的に牽引しよう
 - DXを踏まえてTQMを進化させよう
 - ソフトウェア品質技術を進化させよう
 - アクセルになった事例を皆で生み出し共有しよう
 - 前例踏襲型QA部長やヒラメ型QA部長はもういない
 - インチキなコンサルタントや自称専門家、偉そうにするだけで役に立たない大学の先生を適切に批判し、その情報を共有しよう
 - 自分で何もしないで「世界に冠たる品質立国」とかほざくのはもう止めよう
 - 日本にいてだけで何もしないで勝手に品質が上がるほど、品質は甘くない



最後に: 品質とは – どれでしょう？

- 生産工程における不良品の数の少なさ
- 寿命の長さや故障の少なさ
- 設計工程における設計ミス数の少なさ
- どのくらい規格を満たしているか
- 顧客の要求に対する合致度
- 当たり前品質と魅力品質
- SDGsや公平性などの社会的よさ
- 品質の高いものを生み出す組織能力
- ユーザにとっての価値
- (ワインバーグ曰く) 誰かにとっての価値

品質の「品」は
品物ではなくて
品格を意味しています



答え：全部です

- 石川馨の定義
 - 仕事の質, サービスの質, 情報の質, 工程の質, 部門の質, 人の質, システムの質, 会社の質など, これら全てを含めた「質」
 - SQuBOK1.1.1.7/品質の定義: 石川馨
- 日本的品質管理は「品質とは」という深遠な問いに答えるために、様々な進化をしてきた
 - 源流管理
 - 生産のある工程で不良が出たら、より上流の工程にさかのぼって原因を見つけ解決すべし
 - 出口検査主義からプロセス主義へ
 - 最終工程で不良品を弾くことによる見かけの品質の向上から、工程内でつくりかたをカイゼンする真の品質の向上へ
 - 設計品質・企画品質
 - 生産がかなり安定したので設計の品質や製品企画の質に踏み込むようになった
 - 統計という超強力な道具が使えなくなった
 - 信頼性工学との融合
 - 元々別の学問だった信頼性工学と品質管理は密接に連携するようになった
 - 規格の適合度・顧客の要求に対する合致度・顧客満足度
 - 規格が要求することやお客様が要求すること、「satisfy」されることは、果たして我々が目指すべき品質なのか？
 - 規格が要求することを楽々クリアしなければ技術が進化せず、いつか競争に負けてしまう
 - Customer satisfactionではなく、Customer delightを目指すべきである
 - 当たり前品質・魅力品質
 - 狩野モデルで提唱されているが、魅力とは何かにまで踏み込んではいない
 - 官能特性・感性品質
 - 人間の感覚器官やセンスで評価される品質に踏み込むようになった



品質？価値？魅力？

- (少なくともソフトウェア開発において)(多くのサプライヤにおいて)
魅力や価値についてともしっかり考えているエンジニアはあまり見聞きしない
 - UXとか叫んでいる割に、単にユーザビリティの言い換えに過ぎない場合もある
 - アフォーダンスに従えばUXが高まるのだろうか
 - 機能的価値、情緒的価値、自己表現的価値
 - 機能的価値:「ボルボは市販車で初めてサイドエアバッグを装備した」/
情緒的価値:「ボルボは安全である」/ 自己表現的価値:「ボルボに乗る自分は家族思いだ」
 - ハーレーに乗っている人はマルボロを吸ってバーボンを飲み革ジャンを着て入れ墨を入れる？
 - 経験価値、ストーリー価値
 - テディベアを買う小さい女の子はどんな価値を手に入れている？
 - ミッキーマウスは世界中のディズニーランドで1匹しかいない？
 - グッズドミナントロジック、サービスドミナントロジック
 - GDLは価値を全て提供側がつくるが、SDLは顧客と共創する
 - 優れたテーマパークは設備トラブルさえもエンタメにする
 - 社会的価値まで考える必要がでてきた
 - SDGs、カーボンニュートラル、EV
 - 「誰かにとっての価値である」と叫んでいる人の多くは単に顧客価値だけを意味していると誤解している
 - あれは「ロールによって捉えている品質は違うよ」という話にすぎない
 - 価値が変わると品質保証手段も変わる
 - 自動車におけるシフトライトとサブスクリプション



品質とは

- 品質は、創造されなくてはならない
 - お客様が欲しいものはどんどん変化し、お客様自身ですら事前に分からないかもしれないので、お客様以上にお客様のことを考え抜いてお客様が欲しくなるモノコトを創造する必要がある
 - そのために組織全体が賢くなくてはならず、伝統と進化がずっと必要になる
 - ソフトウェアの質から、製品やサービスの質、事業の質、経営の質に拡げていく必要がある
- 品質とは何か、という深淵なる概念は、ずっと問い直し続けなくてはならない
 - 自分たちは何を指すべきなのか、
自分たちによってお客様や社会はどのように変化しシアワセになるのか、
そして自分たちはどうあるべきなのか、
どこを目指すべきなのか、
を継続的に問い直し続けることが必要である
 - 企業や事業、製品としての「自分たち」と、QAとしての「自分たち」のあり方を両方考え、戦略に落とし込み、実行しなければならない



A man with a tattooed arm is standing on a ladder, painting a large mural on a brick wall. The mural features a large, stylized red and white shape, possibly a flame or a character. The wall is made of dark grey bricks, and there are other murals visible in the background, including one with a circular frame. The scene is outdoors, and the lighting suggests it might be late afternoon or early morning.

(ソフトウェア)品質保証は
組織を賢くするために
あらゆることをする
最高にやりがいのある
クリエイティブな仕事である

私の周りでの研究開発の例: QMファンネルと品質文化

SEA-SIGSQAでは3つ合わせてQM(品質マネジメント)と呼んでます

- QMファンネル(3D版)のそれぞれの側面はスペシャリティ(得意技)を表す
 - TE - テストやレビュー、マトリクスの測定など製品やサービスの評価技術のエキスパート
 - 検証技術と価値重視文化を担う
 - PE - 様々な自動化を行うエキスパート(SETやSRE)
 - 自動化・デジタル化技術とエンジニアリング文化を担う
 - QA - 組織能力を高めるエキスパート
 - 組織能力向上の技術と文化を担う
- スペシャリティをきちんと区別しないと、適切にスキルを高められない
 - スペシャリティのバランスが悪いと、まっすぐ力強く品質向上していかない
 - その一方で、それぞれのスペシャリティをサイロ化させずに融合していかなくてはならない
 - それぞれのスペシャリティが別の人に割り当てられている状況でも、他のスペシャリティのことを理解していなければ良い仕事はできない



p.16

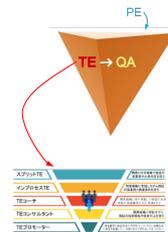
© NISHI, Yasuharu

<https://www.slideshare.net/YasuharuNishi/quality-management-funnel-3d-how-to-organize-garelated-roles-and-specialties>



ソフトウェア開発における品質文化

- 品質文化の土台となる原則には色々たくさんある
 - イマドキのソフトウェア/デジタルな組織でよく言われる原則
 - 「質とスピード」
 - 価値を届けるとかビルドトラップとか
 - 技術的負債とか木こりのジレンマとか
 - 自動化とかパイプライン化とか
 - ホリスティックテストイングとかシフトレフトとか
- どうも趣きの違う原則が含まれている気がするぞ...
 - 価値を届けるとかビルドトラップとか
 - 技術的負債とか木こりのジレンマとか
 - 自動化とかパイプライン化とか
- 「品質文化」と一括りにするのはちよと大きすぎる気がする
 - サブ文化に分けてみると見通しがよくなる
 - 「価値を届けるとかビルドトラップとか」 → 価値文化
 - 「技術的負債とか木こりのジレンマとか」 → 組織能力文化
 - 「自動化とかパイプライン化とか」 → エンジニアリング文化
 - ドメインや組織の構造や状況、歴史などによってサブ文化間のバランスが異なる
 - スケールが重要な産業においてはエンジニアリング文化に傾けた方がよいし、組織が大きくなっていく時期には組織能力文化に傾けた方がよいかもしれない
 - 3つのサブ文化は独立ではなく、一つの品質文化を構成する3つの側面なので、互いに良い影響を与え合うようにしなければならない
 - 「設計を改善するのはいいが納期は絶対だ!」のような、サブ文化同士をトレードオフにしてしまう原則は採用してはいけない

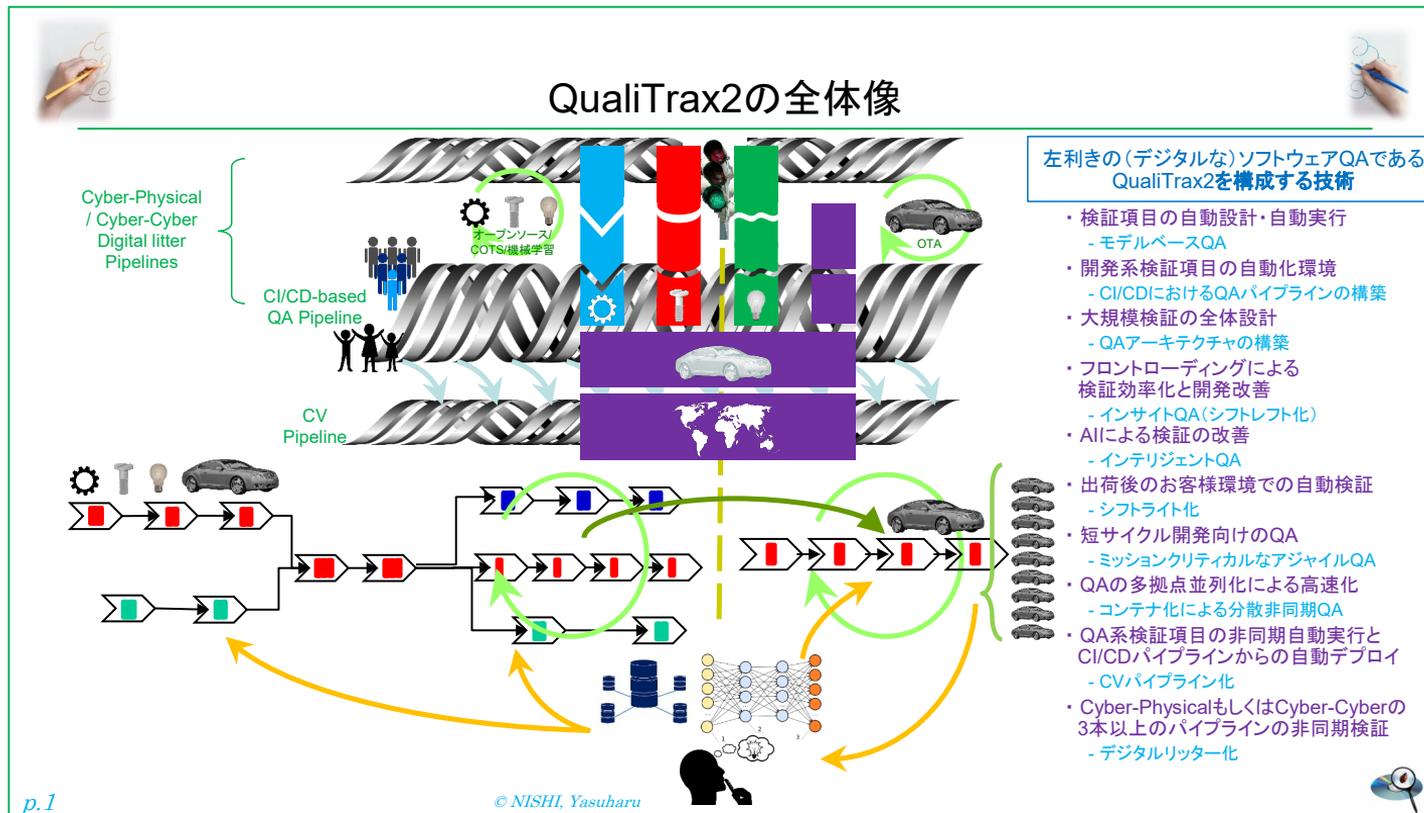


p.26

© NISHI, Yasuharu



私の周りでの研究開発の例: QualiTrax2

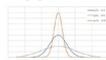


私の周りでの研究開発の例：ロジックキューブ

ばらつきとロジック

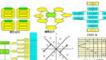
- 大量生産工程では、ものごとを正確に捉えるには統計を使うことがポイントになる

- 種かにQC7つ道具は統計的な道具の集まりに見える
 - チェックシート、グラフ、散布図、ヒートマップ、管理図、特性要因図、層別！？
- 統計的に品質をカイゼンするには、分布のばらつきを減らせばよいという考え方になる
 - 不良品は要するに、分布の中心から一定以上遠いもの、だからである
 - 性能を上げる(分布の中心を移動させる)ことは品質のカイゼンではないことに注意する
 - エンジンの一気量を上昇させた、たぶんすべて性能のエンジンが欲しいには別物の箱でも
 - 天才級ローが体系化した品質工学(9sigma)は条件のばらつきに着目した



- 一品もので言語データの多い工程では、ものごとを正確に捉えるにはロジック(論理)がポイントになる

- 設計や企画(ソフトウェアの開発)といった一品もので言語データの多い品質管理には統計が使えなくなった
 - というわけで、企業によってTQC/TQMが統計部門にすぶる資料の多いことがままある
 - TQCはそれに対して新QC7つ道具などを開発したが、まあそんな...
- きちんとしたロジックを構築する工夫(道具立て)は他にもある



- 事業に基づく管理
 - とか品質管理が絡むと分かってないヤツが標準でものを言うことが多いので事業に基づく管理がとても重要になる
 - 5ゲン主義 = 3環 + 2原
 - 現場、現物、現業、原理、原則

p.14

© NISHI, Yasuharu

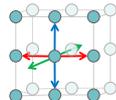
ロジックキューブを大きくしていく

- ロジックには3方向ある
 - 縦 → 抽象 - 具体
 - 横 → 目的 - 手段 / 結果 - 原因
 - 奥 → 全体 - 部分
 - 部品 - 集合部品 - 製品 - システム - サービス - エコシステム



- このロジックをつなげて広げていくときちんとしたロジックが構築できる

- ロジックキューブを広げてつなげて見えるように思考していく
 - 解像度が上がる、と実感する人いる
 - 先入観や思い込み、勘定を捨てることができるようになる
 - 「目的指向」が身につくので仕事の質が上がる
 - 開発エンジニアと品質エンジニアでロジックの方向性に差があると補完できる
- 行ったり来たたりしながら思考する
 - 思案しながら矢印の始点と終点を繰り返して往復できるようにする
 - そのうち、ちょうどいい始点度やちょうどいい原因、ちょうどいい範囲が体感できるようになる
- 組織の場合の例
 - 縦: 経営 - 現場 / 横: 上工程 - 下工程 / 奥: 個人 - チーム - 部門 - 企業 - 社会



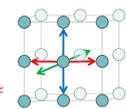
p.15

© NISHI, Yasuharu

ロジックキューブをつなげて大きくしていく

- ロジックキューブをつなげて大きくしていくという品質管理の原則の例

- Why five(なぜ5回)
 - 普通は2-3回なぜ繰り返すが、それでは表面的にすぎず根本原因まで分析できないよ、という経験則
 - 5回を目的に信じるのは未だ転動
 - 「なぜ？」という質問に対する回答者の解釈に脱線になるとよい原因を奪えず、根本原因を導き出せる場合も多い
 - 「なぜか」それで何が嬉しいのか」という問いを繰り返すことは、脱線に囚われず本質的な問題を捉えられるようになり、目的そのものを新設または管理を本質化する
- 源流管理・品質の作り込み
 - 不具合の原因を上工程までさかのぼってカイゼンすること
 - 不具合が入り込まないよう上工程から配慮・工夫して開発をすること
- 次工程(後工程)はお客様
 - 自分の次工程がやりやすいようにしないと品質は上がらないという教訓
- 水平展開(横展開)による未然防止
 - 同じメカニズムで発生する、まだ起きていない不具合を予測できるようにすること
- 目的指向
 - 目的を達成できる手段は他に無いかつも探し続ける
 - 手段を目的にないで、現状の品質向上に意があると感じ取った瞬間に別の手段を仕込み始め、十分ではないものも少しずつイケルと思った段階でリスクを取って乗り換えることができる
 - 目的を長期的かつ大域的かつ合理的な案に落とし込み、泥臭く愚直に施策をやりきる継続は強い



p.16

© NISHI, Yasuharu



私の周りでの研究開発の例：技術ロジスティックス

良さの知識と悪さの知識 (ポジティブナレッジとネガティブナレッジ)

- 悪いものは、良くないものではない
 - レビューのジレンマ
 - バグを無くすためにしっかりしたレビューをやろうとすると、チェックリストがどんどん分厚くなり、レビュー対象範囲もまったく絞り込めなくなるし、それでもトラブルが起きるとダブルチェックやトリプルチェックをするようになっていく
 - 「何かが足りない」「何か欠けている」「何か間違っている」というタイプの品質管理を行っている、このジレンマが発生する
 - 品質管理では「悪さの知識(ネガティブナレッジ)」がとても重要である
 - "To err is human"(過つは人の常)
 - 開発者によっては、製品によっては、製品ファミリによっては、ドメインによっては、国によっては、誤りやすいパターンが必ずあるので、そこを狙えば効率的にカイゼンできる
 - 例) 仕様書で備考の多い部分にはバグが多い
 - 悪さの知識のないFMEA(故障モード影響解析)は膨大な作業で疲弊してしまう
 - 悪さの知識に注目するのは日本独特なのかもしれない
 - ドイツは「もっとよいものをつくるマインド」、日本は「悪いところをどンドン潰す」マインド
 - ゴールが無くてもカイゼンを進められる希有な国民性
- 「これでよい」と言える品質管理も同じくらい重要
 - 経験の少ない医師は病気がばり診ているので健康な臓器写真を判別できなくなってしまうらしい
 - 品質管理をしているとどうしても不具合のリスクばかり目に付く、そればかりでは開発から信頼を失ってしまう



p.11

© NISHI, Yasuharu

技術ロジスティックス

- 組織に「技術ロジスティックス」を構築しカイゼンし続けることで組織の競争力を向上し続ける
 - 技術ロジスティックスとは、技術の知恵が組織内(と外)を循環し、製品の開発などにガンガン使われ、どんどんよい知恵が増えていく仕組みである
 - 動脈系(ポジティブナレッジ)の技術ロジスティックスが無いと、エンジニアは現場でベストでないプラクティスを使わざるを得ず、生産性や品質を落とすことになる
 - 静脈系(ネガティブナレッジ)の技術ロジスティックスが無いと、エンジニアが行った工夫を技術化して蓄積して再利用できないため、仕事やりっぱなしになり賢くない
 - 技術ロジスティックスの構築やカイゼンはどうしても後回しになるので、意識して投資し時間や優秀な人材を割り当てる必要がある
 - 技術ロジスティックスが負荷な現場は労働力で戦わざるを得ず、頑張っても技術が高まらず、どんどん疲弊していく
 - プロジェクトの「外」でやろうとするのは正しいロジスティックスの姿ではない
- 技術ロジスティックスに必要なもの:
 - ポジティブナレッジだけでなくネガティブナレッジ
 - バッドノウハウではなくグッドノウハウ
 - バッドノウハウ: 知らないが開発できないが本来知る必要のない制約事項(高林哲)
 - グッドノウハウ: 本質的でセンスのよい汎用的な技術や考え方
 - 自分たちの技術がグッドノウハウかバッドノウハウかを判別し、グッドノウハウを育てる文化を醸成する
 - 仕事を覚えるということがバッドノウハウの蓄積を意味している組織は案に多い
 - バッドノウハウにまみれたエンジニアは潰しがきかず、組織が硬直する
 - テックシェア(技術の棚)を作って整理する
 - 適切な抽象化・パターン化や具体化の事例・ガイドをつくり、経験や情報を知恵化する
 - ナレッジが循環する仕組みをつくり、循環しナレッジが増えることにインセンティブを与えるような仕組みも同時につくる



p.12

© NISHI, Yasuharu



私の周りでの研究開発の例:ソフトウェアトラップ

ソフトウェアトラップ分析(不具合モード分析)

- ソフトウェアトラップ(不具合モード)を含む仕様書や母体設計書から作られる構造や機能にはバグが多いと推測する技法
 - 開発者がどこをどう間違えるか、に着目する技法
 - ソフトウェアの入力成果物(仕様書や母体設計書)に開発者がバグを作り込んでしまう「トラップ(罠)」が含まれているため、開発者はそのトラップに引っかかってしまい吸い込まれるように出力成果物(設計書やソースコード)にバグを作り込んでしまう、という考え方である
 - 例)「優先順位の表現で1(高い)~5(低い)と5(高い)~1(低い)が混在すると混同しやすい)
 - 河野哲也氏(現メルカリ)らが不具合モード分析として確立し、嬉野綾氏(ワークスアプリケーションズ)らがバグシエルジェ(ASTER)という研究グループで議論を進めている
 - 類似品に注意: 水平展開やフロントローディングを行えない類似品もある



p.86

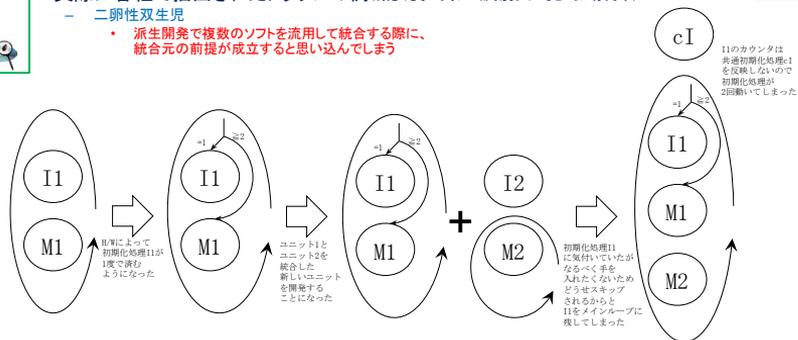
© NISHI, Yasuharu

ソフトウェアトラップの例



実際に各社で抽出されたトラップの例(特定できないように一部改変しているところがあります)

- 二卵性双生児
 - 派生開発で複数のソフトを流用して統合する際に、統合元の前提が成立すると思い込んでしまう



p.36

© NISHI, Yasuharu

