


ソフトウェア品質温故知新

松原 友夫
 松原コンサルティング
 IEEE Software産業諮問委員会委員
 Cutter IT Journal編集委員会委員
tmatsu@xa2.so-net.ne.jp


Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 1



目次

- 故(ふる)きを温(たず)ねる
 - ITの夜明け
 - 機械工場にて
 - コンピュータ工場にて
 - 故きを温ねて考えたこと
- ソフトウェア企業での展開
 - 学んだことを利用した仕組み作りのいろいろ
 - 影響を受けた人と思
 - 私のアプローチ
- 組織文化について


Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 2



ITの夜明け(1947-56)

- 大学(機械科)には入ったけれど
 - 昼間は大学、夜はアルバイトとして米軍極東空軍基地で夜勤
 - キーパンチャーからオペレーター、最後はスーパーバイザー
 - 個人別エラー統計の表示と個人評価
 - 訓練された状態でのエラー率は0.3%
 - 後に、配線エラー率、原稿の誤字率がほぼ同じであることを知った
 - 次々と新しいIBMのパンチカードマシンが導入された
 - 会計機(Accounting Machine)、計算機(Calculator)、照合機(Collater)、分類機(Sorter) etc.
 - ENIACのようなバッチボード式プログラミング
 - こんだもので偏微分方程式を解いたりエンジン寿命の計算
 - バグよりもデータ品質が問題だった
 - きつい制限の中で工夫する面白さを学んだ
 - 兵站のシステム構築と運用を任された
 - 新しい機械が来ると最初に使いにされたのは日本人
 - アジア各地の他の補給基地と比べて格段に日本のレポートの質は高かった
 - 多くを任せられ胸を張って仕事をした


Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 3



工場を挙げての改善活動に参加(1956-65)

- 当時の日立電有工場
 - 都内最大の敷地を持つ機械工場
 - 製品は、ポンプ、クレーン、ブルドーザーなど
 - 機械屋と工具からなるコミュニティー
 - 製品の納期遅れ、コスト高、品質問題に振り回されていた
 - 個別の遅れを取り戻そうとして特急を乱発し混乱を招く
 - 製品移管に伴う
- 工場を挙げての改善活動を開始
 - 改善哲学を持つ工場長のリーダーシップで
 - 工場長直属の企画室が推進
 - バックグラウンドにこだわらない人材配置
 - 技術屋がファイリング、帳票設計主導
 - 法科出身の資材屋がヴァリュウ・アナリシス主導
 - 各部門に合理主義者が揃っていた
 - 職位や年齢に関係なくよい提案はどんどん採用した


Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 4



工場を挙げての改善活動に参加(1956-65)

- 工場の文化
 - 機械屋文化
 - 徹底した現場主義、実証主義
 - 現場、実物、実験結果の観察から発想
 - 実験と試作は必須
 - 計数、グラフによる観察や制御はway of life
 - エントロピーを下げようとする
 - 自動制御理論に慣れ親しんでいる
 - 事実に教えられる謙虚で誠実な人々
 - 設計部と製造部との間に緊密な連帯感
 - 試作では設計は製造に頼り設計問題では製造は設計に頼る
 - 検査部長には工場長を超えた出荷承認権限があった
 - 前例に拘らない改善策の実施(後述)
 - 亀の機械化は岩波新書「事務のオートメーション」に紹介された
 - 寛大な人材放出 人は真いがかかったときが華
 - このお陰で私はコンピュータ事業部に転出できた

Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 5



工場を挙げての改善活動に参加(1956-65)

- 改善哲学をいくつか
 - 性善説を前提にしないと改善はできない
 - ただし違反者には厳罰
 - ゆとりの法則: 一升まずには一升
 - 特急の乱発を戒め工場内を整理整頓
 - 根本原因の追及: 下痢をしない改善は本物ではない
 - 失敗に対して寛容
- 設計・製造の改善
 - リベット接合、溶接 食い溶接機による自動溶断/溶接
 - 作業の安全と品質の向上
 - 板金工場の作業員全員がNCプログラマーに変身
 - ポンプのケーシングを鋳物から厚板へ
 - 主な不良原因である鋳物利用を減らすための設計改善
 - 設計計算にコンピュータ利用
 - 技術計算委員会がコンピュータ利用を促進
 - ポンプの異型計算、水撃現象数値シミュレーションなどをIBM 650で
 - マシン語、最適化アセンブラSOAP, FortranSIT, Fortranの経過を体験

Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 6

工場を挙げての改善活動に参加(1956-65)

- 生産システムの改善
 - 材料、部品、作業量のデータベースで生産計画を合理化するためすべての伝票を穴あきカード(パンチカード)に変えた
 - カードを現場に配る前に複製し、一方をマシン室に保管し、作業が完了すると、原価計算の入力と同時に保管したファイルと照合して除去する
 - こうして維持するステータス・ファイルから毎週作業量を予測し作業計画を立てる
 - カードのアカウントNo、金額、期限、貸借コードの欄を合わせたので締め切りの翌日には決算できる
 - 現在でさえこれできていない企業が多い
 - 期限を暦日から遡へ
 - 53枚の週めくりカレンダーをあらゆるところに掲示
 - 運搬工の廃止
 - フォークリフト運転手が直接それを行へるように環境を改善
- 創造的でエキサイティングな日々だった
 - このシステムは他の工場に広がった

コンピュータの利用者から開発側へ(1965-70)

- 初仕事はデスマーチ・プロジェクト
 - 科学用大型コンピュータで税務システム
 - アプリケーションと並行して同じチームで事務用OSを開発
- 初代科学用パッケージ開発課長
 - アンバンドリングの走り
 - 初のアメリカへのアウトソーシング
- 兼務でプロセス制御プロジェクトまとめ
 - 汎用機でプロセス制御(電力負荷配分システム)
 - 専用OSとアプリケーションを開発
 - 稼働後はバグゼロ
 - オイルショックで予想外の大活躍
- 方式設計部で大規模複雑なシステムまとめ
 - 都市銀行オンラインシステム総責任者
 - タスクはメーカーとしてのシステムの稼働保証
 - よるす苦痛、トラブル承り
 - トラブル解決時は裁判官

ユーザーからメーカーへ(1965-70)

- 方式設計部で複雑系のシステムへの取り組みを経験
 - みどりの窓口がオンラインシステム技術の源流
 - 都市銀行1次オンライン開発プロジェクトの総責任者
 - 最大の技術問題は処理能力とシステム稼働率
 - そのために機能を優先度の低い機能を大胆に切った
 - このために特注OSを開発しコンピュータを大幅改造
 - システム障害時の被害を減らし回復を早めるため
 - 三段構えの進行処理
 - 完全停止でも手作業で払い出しできるようにした
 - Y2K問題には無縁の基本設計
 - 方式設計部には優れたシステム屋があり、ここから優れたシステム屋が育った
 - ここで言うシステム屋とは、複雑大規模なシステム全体を設計でき、理解でき、問題指摘や対策ができる人と言う
 - システム屋は複数の異なる分野での経験によって育つ

ユーザーからメーカーへ(1965-70)

- コンピュータ生産はデザインオートメーションシステム無しにはできない
 - 当時の最速の大型コンピュータをふんだんに用いた
 - 論理設計から実装設計までのファイルを統合
 - すべての設計者の机上に端末機が置かれ、設計作業のほとんどは膨大なDAツールで行われた
 - これは初めての開発環境と呼べるもので、設計者の能力評価や指導にも使われた
 - 技術進歩に追従するために常に膨大なソフトウェアを改造・開発した
 - 極めて早い時期にDAとCIMを統合した例である
 - 後にSEMATECを訪問し、「いかにして生産にソフトウェア思考を導入するか」というテーマで討論したときに、この例を発表した

故きを温ねて学んだこと(1)

- エンジニアリングの取り組みはソフトウェアでも適用できる
 - ソフトウェアにも辛い社名がソフトウェア・エンジニアリングだった
 - ソフトウェア・エンジニアリングを学ぼう
 - 常に刺激を受ける場に身を置こう
 - マトリックスを文化にしよう
- 問題発見、解決指向
 - 既存モデルにこだわらない
 - 機械工場、コンピュータ工場で作ったことで使えそうなことは何でもやってみよう
- 針灸方式の優先順位づけ
 - Root causeを見抜きそれから解決すれば少ない努力で改善される
 - それにはシステム思考が有効
- 技術者の意識改革、スキル向上、自立を最優先
 - 仕組みを用いて
 - 我慢する
- プロジェクトの経過と実績は原則的に記録に残す
 - 知識ベースを作る
 - 技術移転のため

故きを温ねて学んだこと(2)

- ハードウェアとソフトウェアが大きく違う
 - 複雑さの程度 (Complexity)
 - 機械図面のチェックは勘所を見ればよいので瞬時にできる
 - ソフトウェアのコード審査は、書くのと同じくらい時間がかかる
 - 修正フィードバックループの長さ
 - 機械部品の製造では加工不良は原則的にその場で直す
 - ソフトウェアでは作った本人さえ気づかず次のプロセスに送ることが多い
 - 不良品の比率 (Yield)
 - 機械では不良品は伝票を起こして再製するから仕損費として正確に把握できる
 - ソフトウェアでそれをやったら仕事にならない
 - ツールの効果が現れる時間 (Learning Curve)
 - 機械では、品質、生産性、コストなどの改善で新鋭機械を導入することが多いが、操作に慣れればすぐに効果が現れる
 - ソフトウェアのツールは使いこなすのに長い年月がかかる
 - 技術移転 (Technology Transfer)
 - 機械では「モノ」を通して技術移転ができる
 - ソフトウェアでは人を通して技術移転が行われる

ソフトウェア会社での展開(1970-91)

- 設立当時(1970)の状態
 - プロジェクトはトラバだらけ
 - このままでは会社の経営が成り立たないという危機感
 - 最年長は経験2年のプログラマ
 - プロジェクト管理経験皆無
- いままでの経験と智恵を生かす時が来た
 - 仕組みを使う: 最優先はコスト意識を高めること
 - 数字とグラフを使う: e.g. メトリクス文化に
 - 自動制御を応用する: e.g. グラフを使おう
 - 現場主義 実証主義
- まずは基盤作りから
 - 第一線管理者の自立心を育てる
 - 「管理者は我慢せよ」、「小さな火傷をさせよ」
 - プロジェクト管理教育を最優先に
 - プロジェクト管理の模擬練習
 - 合格すれば請負へ
 - 図によるプロジェクト管理(次ページ)

意識改革に図を用いた

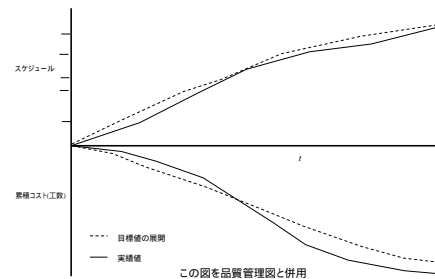
- 経験2年のプログラマにコスト意識を植え付けるのが急務だった
 - 実態をそのまま知らしめることで判断力を養う
 - 育つまではじっと我慢する
 - 敢えてコストを表に出さず、プロジェクト目標の優先順位を「納期」「品質」「コスト」とした
 - 納期を最優先にしたのは顧客に迷惑を掛けないことを第一としたため
 - 品質を制御できればコストも制御できると考えコストより優先した
 - 彼ら全員をプロジェクトに責任を持つ「主任」に格上げ
 - 管理ツールとして、2つの図を与え彼らにプロットさせた
 - 原価工程管理図: 日程とコストはこれで管理する
 - バグ死滅曲線: これはおなじみの日立から継承した図
 - これらの図の簡単な見方を教えた
 - 彼らは現象から学び、優秀なプロジェクト管理者に育った
 - これは機械工場で学んだことの応用の一つ

生きた図の隠れた効用

- 管理ツールとして役に立つ実感が得られるのでメトリクスが定着する
 - 図の曲線パターンの意味を議論するようになればしたもの
 - 自ら学ぶ仕組みなので座学や説教は最小限でよい
- メトリクス文化の浸透が目に見える
 - メトリクス文化とは
 - 図が職場のあちこちに
 - 図が会議で多様される
 - 数字を把握することを厭わないようになる
- 他から良い図を取り入れるようになる
 - 論文は優れた図の宝庫
 - 真似しても文句を言う人はいないし、日本発の図は外国で多く使われている
 - 例えば、バグ死滅曲線、魚の骨チャート
- 図の作り方、使い方が進歩する
 - 問題に適した図法を考えるようになる

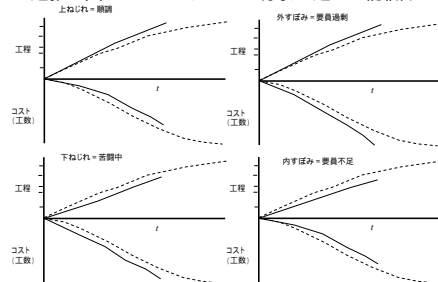
図によるプロジェクト管理(1)

- プロジェクト管理用ツールとして与えた管理図



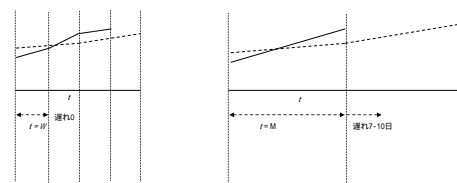
図によるプロジェクト管理(2)

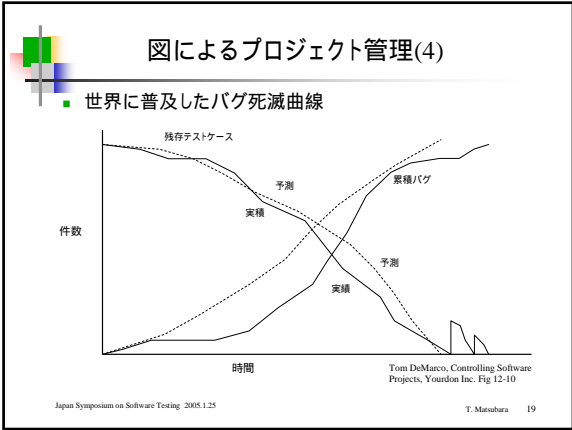
- 進捗を図の4つのパターンで現象の違いを認識する



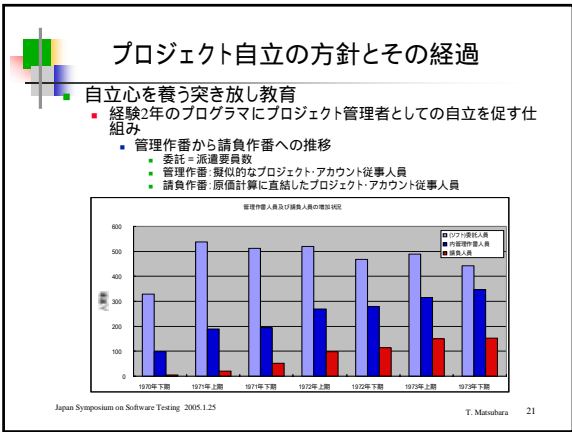
図によるプロジェクト管理(3)

- 制御を可能にするには
 - 目標値の時系列展開と
 - データ間隔 t と時間の遅れをどのように決めるが重要





- ### 生産性・品質監査制度
- 会社設立1年後から開始
 - ハンフリーの最初の本に見られるアセスメントと酷似
 - 固定チームで半年に1度現場を巡回
 - チームは担当役員、生産技術部部长、生産技術部員からなる
 - 同じ人が巡回することに意味がある
 - 今で言えばSEPG
 - 現場でやるのは関連資料を随時求められるため
 - 資料は生産性、品質の実績と前回改善要求事項
 - 全社メトリックスシステムで準備
 - 監査というより助言と技術移転援助
 - 現場と合意した評価と次回改善要求事項を報告
 - 当時はプロダクトとプロセスを意識的に分けていなかった
 - プロダクト指向が強かったように思う
 - 数年後に品質監査を分離
- Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 20



- ### エントロピーを減らす
- エントロピー増大の法則:
 - エントロピーとはある系の無秩序性の程度を表わす物理量で、物系が平衡に向かう過程で発生し、一方的に増大する特性がある。自然状態では、物質は拡散し、エネルギーは低級化し、情報は失われる
 - 機械屋はエントロピーを増大を極度に用心し下げようとする
 - 機械工場では品質低下や納期遅れを防ぐための金言
 - 工場に部品を無理に押し込むな。一升すには一升、
 - 混乱より秩序。整々と運らせよ。
 - ソフトウェアでのエントロピー低減
 - 情報隠蔽やカプセル化を適用してより透明なシステム設計またはプログラム構造にする
 - Separation of Concern, Modular Conceptなどの適用
 - 参考: D. Parnasの情報隠蔽(Information Hiding)、N. Wirthのカプセル化(Encapsulation)
 - 文書やソースコードなどの派生物の様式を揃える
 - 個人のプロセスの質を揃える
 - 仕様の際限のない膨張を抑える
- Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 22

- ### 治工具開発をヒントにしたツール開発
- 機械工場での治工具
 - 製造に必須
 - 公式に図面を書いて製作
 - 費用はそれを使う数件のプロジェクトで負担
 - ツール開発制度
 - 目的:
 - ツール開発を公にする
 - ツール開発の提案を募集し審査の上開発予算を付ける
 - ツールの技術的水準より利用人口を優先する
 - 実績例
 - 初期に提案されたものは単体テスト用アニメーター
 - 最初はアセンブリ言語、次第にコンパイラも
 - カラレッジメジャーを出力シートに表示し完了基準に
 - 開発したアニメーターのいくつかは商品に
- Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 23

- ### 影響を受けた人と思想(1)
- Manny Lehman, Laszlo Beladyのソフトウェア進化論
 - Program Evolution Process of Software Change, Academic Press 1985
 - E-Type
 - 環境の中に埋め込まれるもの
 - 環境は常に化する
 - 要求仕様は決まらないのが当たり前
 - ここで言う環境とは、ソフトウェアが稼働する周囲の環境のこと
 - S-Type
 - 仕様通りに作れば動くもの
 - 以前はその典型が組み込み型だった
 - 現在の組み込みは環境に依存するからS-Typeはほとんど存在しない
 - 仕様がダイナミックに変化するの当たり前
 - 動く標的との戦い
- Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 24

影響を受けた人と思想(2)

- Niklaus Wirthのカプセル封じとLilith
 - WirthはAlgol 60, Pascalの開発者として知られる
 - 1983年にスイスのETHに彼を訪ねた時に見せてもらったソフトウェア開発専用ワークステーション”Lilith”で目から鱗
 - ヴィルトが発明したModula-2の中間表現、Mコードのエミュレータ
 - プログラムの多様な表現が49のウィンドウから見られる
 - Modula Conceptに基づいて作られているので高速
 - これを入力しようとしたが権利を与えたアメリカの会社が過剰仕込みで倒産し失敗
 - 当時8ビットPCの時代にこれをまねて分散開発、実機外開発を開始した

Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 25

影響を受けた人と思想(3)

- David Parnasの情報隠蔽
 - 機械屋にとって納得できる理論
 - 複雑なシステムを見通しよく設計する
 - システム全体に関わる情報バスと局所的なバスを明確に区別する
 - 設計でのカプセル封じ
 - 情報隠蔽はOOの基礎理論の一つ
 - バグが取りやすく、メンテがやりやすい
 - 自分が開発した理論は必ず実践するのが彼の主義
 - 1984年初めて会ったとき、情報隠蔽で設計した航空機搭載ソフトウェアの膨大な設計書を贈られた
 - 直言居士としても知られている
 - 大学での教育批判
 - ‘ソフトウェアエンジニアリングは床入りしない結婚’

Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 26

影響を受けた人と思想(4)

- Tom DeMarcoとTim Listerのピープルウェア
 - ソフトウェア開発で最も大切なのは人、ということを最初に書いた
 - プロジェクトの失敗原因のほとんどは人間的要因
 - 人材の採用、教育、チームワーク、人をやる気にさせる、楽しい仕事、など
 - 彼らが私たちが1981年と1984年に書いた論文を本に引用して以来の付き合い
 - Tomとは多くの場面で一緒に活動してきた
 - The Atlantic Guild主催のヨーロッパでのレクチャーツアーに参加
 - IEEE Softwareの委員に引っぱり込んで一緒に仕事をした
 - アメリカ、メイン州Camdenの自宅を数回訪ねた
 - オープンで思いやりのあるコミュニティーを大切にする

Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 27

私のアプローチ

- 現実からの発想
 - 行動するエンジニアリング
- 使えるものは何でも使う
 - 幅広い関心を
 - 自由な発想
 - 与えられたものに拘らない
- 品質は最優先
 - 納期、コストのすべてに波及する
- ツボを押さえる
 - 部分を刺激して細胞を活性化させる鍼灸の原理
 - 少ないエネルギーで目的を達成
- システム的思考
 - 全体のメカニズムを考える
- 優れた組織文化を醸成する

Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 28

優れた組織文化を醸成する(1)

- 文化とは反射的に考え行動する共通意識基盤
 - 一流の運動選手のように
- 雪印乳業の脱脂粉乳中毒事件の疑問(1)
 - なぜ15,000人の中毒患者を出してしまったか
 - 従業員の家族に中毒者がいたはずなのに社内にアラームが上がりなかったのはなぜ？2つの可能性
 - 知っていたが報告しづらい雰囲気があった
 - 製品が低品質なのを知っていて飲むのを避けてきた
 - なぜ厳格なHACCPの承認を受けていながら問題を起したか
 - 規格に従ってプロセスを記録していれば問題箇所を特定できたはずだったのに
 - 牛乳箱に表示したかっただけなので記録をさぼった
 - そのために疑わしい範囲が広がった
 - 不可解な発覚後の対策
 - 人を1,300人減らした
 - 新製品企画グループを新設した
 - 謝り行脚を始めた
 - 別ブランドで赤箱の牛乳を売り出した

Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 29

優れた組織文化を醸成する(2)

- 雪印乳業の脱脂粉乳中毒事件の疑問(2)
 - 私だったら逆のことをやるだろう
 - 品質改善専任のグループを新設するだろう
 - 社内の会議などで製品の牛乳を飲むようにするだろう
 - 消費者へ向けて品質第一のメッセージを発信するだろう
 - 例え、社長、役員、従業員のすべてが自社の牛乳を飲んでるコミニカルを打つ
 - 業績至上主義の企業文化は変わっていない
 - 消費者と欺き信用を失ったという問題の本質がわかっていない
 - 我が家では今も雪印製品は一切買わない
 - 金銭至上主義文化は必ず破滅する
 - ‘牛肉のハンナン’、‘そごう’、‘堤の西武’、etc. を見よ
 - 良い組織文化を醸成するには長い年月がかかるが消えるのは一瞬
 - エントロピーと同じで文化は放っておけば低きに流れる
 - 組織文化を良い方向に持っていくのは挑戦のしがいの面白い仕事
 - ここにもシステム的思考が要る

Japan Symposium on Software Testing 2005.1.25 T. Matsuhara 30

品質を保つ決め手は優れた組織文化(1)

- 優れた文化の条件
 - 組織の上下左右に情報断絶が少ない
 - オープンな雰囲気、オープンな議論
 - 風通しが良い
 - 権威を振りかざす人がいない
 - 新しいことに挑戦する雰囲気がある
 - 気配りが出来ている
 - 品質ではこれが重要
 - プロ意識の裏付けがないと技術的気配りはできない
 - 因習、不文律が少ない
 - 例えば能力があれば出身に拘らずに抜擢される
 - 悪い例： 大学出身を出て、××支店長、工場を経由しないと偉くなれない
 - ソフトウェア開発では
 - 人を大事にする
 - トップがソフトウェアの特性を理解している
 - トップが品質改善のためにコミットしている
 - 開発者が働きやすいように気を配る

品質を保つ決め手は優れた組織文化(2)

- 大きな誤解
 - 規格やモデルに従えば品質は良くなるのか？
 - ISO 9000の認証、CMM(1)の高レベル達成で品質は安心か？
 - トップダウンで支持すれば品質は良くなるのか？
 - 品質管理を教育すればいいのか？
 - QA部門が強ければ品質は良くなるのか？
 - テスト技術が向上すれば品質は良くなるのか？
 - こうしたことはきっかけに過ぎない
- 良い組織文化の支えがなければ高品質は維持できない
 - 人や環境が変わればすぐに元の本阿弥
 - 元へ戻するような改善はやらない方がいい
 - なぜならすべてが労力とコストの無駄になるからだ