

「ソフトウェア品質保証部長の会」からの情報発信

しくみからのブレークスルー
ソフトウェアマイスターと今後の品質保証

2012 ソフトウェア品質保証部長の会 3G

アヴァシス(株)	江口 達夫	(株)ネクストジェン	齊田 奈緒子
ブリヂストンソフトウェア(株)	村野 耕一	(株)日立製作所	梯 雅人
永山コンピュータサービス(株)	向山 正秀	三菱電機(株)	安藤 洋治
永山コンピュータサービス(株)	頼富 宏平	東芝ソリューション(株)	川原 章義
(株)インテック	遠藤 健史	富士通(株)	小林 理一郎
(株)ニコンシステム	千綿 洋一		

はじめに

ソフトウェアの品質向上は、日本においてはこの20年の間、ISO9001やCMMI[®]に代表される「しくみ」に注力がされている。

しかし、

しくみがあってもなかなか高品質のソフトウェアは出来なかったり、しくみがなくても「達人」レベルの人が開発したソフトウェアは高品質なことも多い。

ソフトウェアは、基本は論理の世界であり、システムとしての自由度の高さ、論理の集合であり人に帰着している部分が多いということが原因と考えられる。

我々は、品質保証技術者、ソフトウェア技術者達人と言われるような技術者の「人間像」や「能力」がいかにして「育まれてきたか」を調査／分析し、高品質なソフトウェアにむけての必要な事項を討議してきた。

目次

ソフトウェア開発における日本の品質保証の実態

- ・しくみがあってもいまいち
- ・人だのみだが、質が不足

達人を増やそう

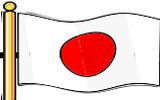
- ・達人は いかにして育まれてきたのか？
- ・達人を 育てるためには？

達人のもっている技術力、考え方などの エッセンスを品質保証のしくみの中で活かそう

- ・達人の技術力／行動／考え方
- ・どのプロセスにどう活かすのか

ソフトウェア開発における日本の品質保証の実態

➤ 米国生まれのソフトウェア工学をそのまま利用しても日本人はうまくいくのだろうか？



	日本	米国
	Software Craftsmanship ソフトウェア職人気質	Software Engineering ソフトウェア工学
ルール・責任	× *1	○
システム／ツール	△ *2	○
顧客や品質に対する意識	◎ *3	×

- *1: ルールに基づき行動する、責任と権限を明確して役割分担に基づいて行動することが苦手で、ルールは形骸化し責任と権限が曖昧になりがち
- *2: システムの構築やツールの導入はリーダーシップを発揮するものがあるとき成功するが、リーダー不在の状況ではシステムやルールは組織的なパワーにならない
- *3: 顧客に対する品質の責任や品質を心配する技術者一人一人の意識が強い。その意識が効果的に作用すると製品の向上につながる

ソフトウェアの品質保証はプロセス主体

➤ ソフトウェア開発の各プロセスの状態を中心とした品質保証

例) すべての機能のレビューは実施したか？

レビューは、定められた指標どおりに実施したか？

各プロセスで発見した不具合の修正がされたかを確認

➤ 経験した問題や課題は、PDCAをまわすことにより、 各プロセスのチェックリストや観点リストにフィードバック



➤ 各プロセスで使用している、チェックリストや観点リストは 膨大な量となっており、全てを本当に確認するには 多大な時間がかかる

きめ細かさは、日本人のもつ特性



時間とお金をかけて、高品質を確保している？

現実は?? (1)

- ソフト開発はPJ単位で実施。PJ管理はQCDのバランス。
つまり、限られた資源(人、金、時間・・・)の中で、高品質を目指す



- できる人の部分は、おまかせでも まあうまくいっている。
- 資源制約の中での保証であり、レビュー、試験などで顕在化した不具合は修正されるが、潜在的な不具合は中々なくなるならない。
(たまたま、発見できたものだけが、修正される)
- 設計工程での不味さは、試験工程で発見されるが、
 - ・設計がPoorで、修正に時間がかかっている。
 - ・納期の関係から、基本設計まで立ち返っての修正でなく、小手先の修正でおわっている。



人のでき、不出来で品質がきまってしまう部分が多い

現実は?? (2)

➤ できる人は限られている



➤ 限られた人を、レビュー専門で活用

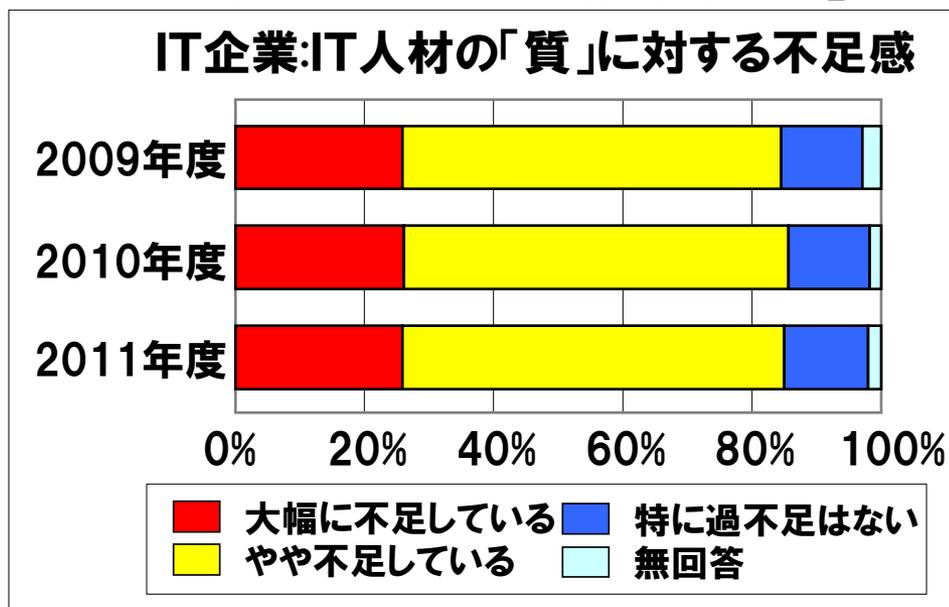
うまく行っている部分はあるが、それでもいくつかのPJは
燃えてしまう

➤ 限られた人を
困難なPJに重点配置
燃えてしまったPJに投入

あげく 体力勝負に・・・



(参考) 出典:IPA「IT人材白書2011」



達人をふやそう

- ・人事採用時は、達人となる素養の人を採用
- ・教育もばっちりやっている(はず)
- ・OJTもしている(はず)
- ・経験もつんできている(はず)

… でも、達人は増えない

- 達人って どんな人だろう？
 達人の人物像は？
- 達人になるためには？
 達人はどうして達人となったのだろうか？
 達人を育てるためには？

達人の人物像（１）・・ こんな人が我々の考える達人です

- 達人は、強い信念、責任感、気概にみちている。
- 達人は、自ら行動するだけでなく、仲間をリードしている。
- 達人は、常に前向きで、先手で考えている。
- 達人は、自分の想いを熱く語っている。
- 達人は、失敗経験を豊富にもっている。
- 達人は、弟子をそだてている。
- 達人は、お客様視点で考えている。
- 達人は、好奇心旺盛で、現状に満足せず向上意欲を持っている。

達人の人物像（２）・・ こんな人が我々の考える達人です

□ 達人は、専門知識 + 幅広い知識をもっている

- ・ 特定分野の専門知識だけでなく、全体的に幅広い知識をもつ

□ 達人は、原理・原則 + 応用力で対応する

- ・ 原理を知っていて、応用もできる

□ 達人は、仕事だけでなく、+ 自己研鑽をしている

- ・ 仕事はきっちりとやりながら、常に自己を高める努力をしている

□ 達人は、自分で解決する + 解決できないところは人を使って解決する

- ・ 人を使える …… その道の専門家を知っている
その道の専門家に、快く引き受けてもらえる

人材採用時には??

各社の人材採用のページを見てみると、

協調性

チームワーク
仲間と協力

信念

責任感、信念、
気概に満ちている

行動力

自ら行動、
仲間をリード

向上意欲

好奇心、素直

➡ **達人になる要素、たっぴいの人材が採用されている(はず)**

参考：
各社のHP

A社:必要としているのは、「次世代を切り拓く」人材。

- ・新しい未来を創造できる人
- ・仲間と協力してよりおおきなことを成し遂げる人
- ・好奇心をもって様々な挑戦ができる人

B社:「N年後の会社を自分がもっと良くするんだ」という気概に満ち、自ら責任をもって日々の仕事に取り組むことのできる人。

C社:自ら考え行動実践し、成果を出すことができる人材

- ・信念を持って行動し、勝戦のできる人材
- ・常に新しいことにチャレンジし続け、周りをリードできる人材
- ・人を愛することのできる人材

D社:・チームワークがとれ、責任感、協調性のある人。

- ・素直に人の意見を受け入れ、将来、成長を期待できる人

E社:新しい技術・サービス・価値の創造に、熱い思いで挑戦する人。

X社:「チームのために」「自分で考え」「行動する」

ソフトウェア開発の達人はどのようにして 育まれてきたか？（1）

達人にヒアリングした結果

□ 入社2～3年で、良い先輩に恵まれた（達人に育てられた）

- ・この時期に教えてもらったことが基礎になっている。
- ・技術的な内容もそうだが、考え方を徹底的に仕込まれた。

□ 他人の作ったプログラムを読んだ

- ・わかりやすいプログラム／わかり難いプログラム、
変更がし易いプログラム／変更がしにくいプログラム
- ⇒プログラムとは、こういう様に作ればよい。こんな作り方をしてはだめを体験

□ 社内の教育だけでなく、自分で勉強した

- ・日経XXなどの、雑誌をよく読んだ。
- ・当時、話題となっていた設計手法などの本を読んだ。
- ・情報処理の資格取得のために、勉強した。

ソフトウェア開発の達人はどのようにして 育まれてきたか？（2）

達人にヒアリングした結果

□ 実際のPJで失敗して、技術、設計の大切さが身についた

- ・ 難解な不具合を解析していくことで、技術的な内容がわかった。
- ・ 運用中のシステムでの不具合の解析時にほしい情報がなく、設計時点でのしかけの重要性を再認識した。

□ 社外の委員会などに参加、その道の専門家として育った。

- ・ 他流試合の場合では、事前に勉強しておかないとついていけない。
- ・ 社内で広めていくためには、重要性などをわかり易く説明するだけでなくPJに入りこんで自ら実践して広めていくことによって、教える側としても迫力がついた。

達人育成のまとめ

仕事の環境を考えて、育てることが重要

自由度を持たせて仕事をさせる

失敗が許容できる仕事をさせる

良い先輩をつけて仕事をさせる

興味のある分野の仕事で、得意分野を伸ばす

達人は、育てるのでなく、勝手に育ってきている面もある

稲盛 和夫氏の言葉にもあるように

人生・仕事の結果＝考え方×熱意×能力

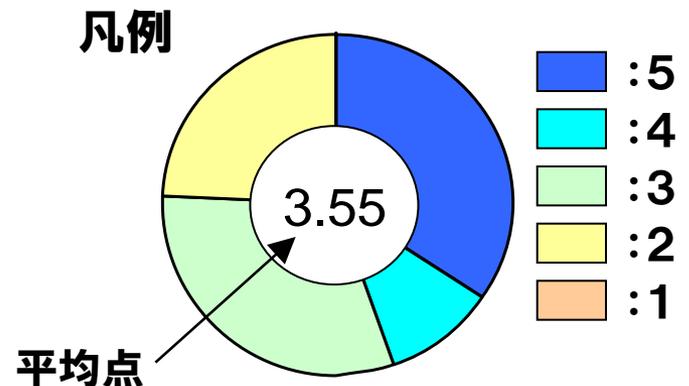
プラス思考で考える、熱意を持たせることが重要



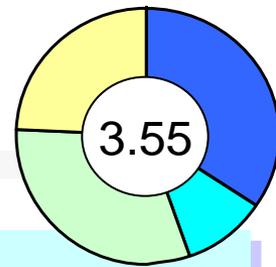
達人のエッセンスを利用しよう

- 達人と一般人の違い
- 達人のエッセンスをどう利用して、品質向上を図るのか？

29人に聞いた
アンケート結果



ソフトウェア開発の達人の設計（1）



□ 達人は、不具合があった時に、解析できるしかけを
予め織り込んだ、設計をしている。

- どの様なデータによる不具合なのかがわかるようなプログラムのなしかけ
- どのような、運用状態なのかがわかるシステムのなしかけ
⇒テスト段階や運用中の、問題が発生したときも、解析が早い

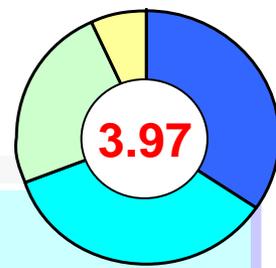
↔ だめな人：問題が発生してから、ログを仕掛ける



●ログの仕掛けをレビューする

- 開発中に必要なもの：不要となったらソースをいじるのでは
ミスを犯すので、パラメータでログを出さなくすることを可能とする
- 運用後の解析に必要なもの：どこで不具合が発生したかだけでなく
どのデータで不具合が発生したか判るようにする

ソフトウェア開発の達人の設計（2）



□ 達人は、非機能 特に性能には気をつかった設計をしている。

- あらかじめ 処理多重化する部分を考えて、多重化している
- DB検索性能を考慮した設計をしておく

↔ **だめな人：性能は 総合試験になって初めて測定、チューニング
(チューニングと称して、一部作り換えもある)**

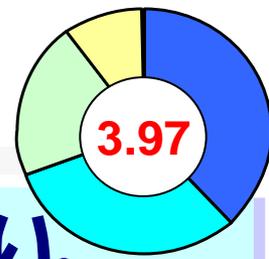


- 性能定義ができているかをレビューする
- 性能をクリアするための設計上の仕掛けがどうなっているかをレビューする
- 性能の確認は
 - 単体試験の段階から性能測定し、無負荷での性能が達成できているか確認する
 - 総合試験段階では、データ量や処理量を加味した性能が達成できているか確認する

性能への対応について(補足)

- **性能要求は、運用条件がなかなか決まらないことが多く、早い段階では簡単には決まらない。**
- **H/Wとの組み合わせで性能実現することもあり、ソフトウェアでは最初からベストの設計をしておくべきである。**
- **さらに達人は、性能要求が厳しそうな部分については、カスタマイズしても、他に影響の無いようなアーキテクチャ設計するなど気を使っている**

ソフトウェア開発の達人の設計（3）



□ 達人は、プログラムに手を加えず再利用する部分が多い。
再利用が多くなるように、パラメータ化設計をしている。

- ・実績あるプログラムを多く使うことは、品質担保の基本
- ・パラメータ化設計により、ソースに手を加えないで再利用できる

↔ **だめな人：何でも自分で作ってしまう。**

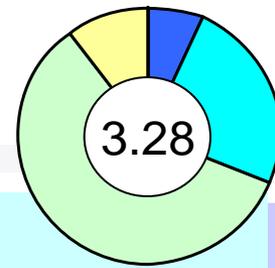
プログラムを流用して、手を加えるが、手を加えたところの確認が甘く、不具合を発生させてしまう。

- **基本設計時に、どのような機能にするのかだけでなく、新しく作る部分、流用して変更する部分、変更しない部分、共通設計をする部分が明確になっているかをレビューする。**
- **パラメータ化すべき部分は、達人の知恵を借いて、明確化し、どんなパラメータにするのかをレビューする。**

「創らない」について(補足)

- 「創らなければ」不具合は新たに入り込まず、ソースはどんどん枯れたものになっていく。
- 究極は1つも「創らず」新たに入り込む不具合もない(使い方による不具合は発生する)状態である。
- 最近のソフトウェア開発においては、上記のような取り組みが強力に推進されている。

ソフトウェア開発の達人の設計（４）



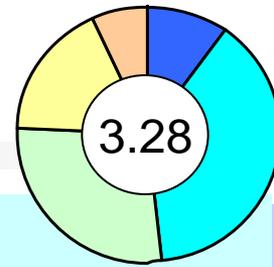
□ 達人は、MW (Middle Ware)などの設定項目は、システム特性や客先のデータ量や運用に合わせて、設定している。

- パラメータ化されている部分は、プログラムだけでない。
- MWもシステムの一部であり、MWのパラメータ設定は、システム設計の重要な項目である。

↔ だめな人：何も考えずに、標準設定(デフォルト)そのまま

- 客先の運用条件が明確になっているのかをレビューする。
- 運用条件とシステム特性に合わせたMWの項目設定をパターン化し、なるべくパターンから選ぶ。
パターンに合わないときも、なぜその設定内容でよいのかをレビューする。

ソフトウェア開発の達人の設計（5）



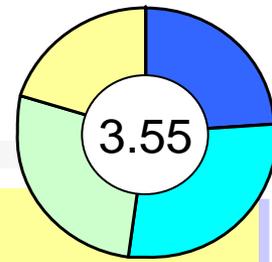
□ 達人は、新技術の採用に関しては、慎重である。

- ・ 新技術を利用する時は、予め検証して利用できる部分、利用できない部分を切り分けている。

↔ **だめな人**:生産性があがると、何も考えずに利用し、
問題が発生してから、あたふたする。

- 新技術を利用する際は、
事前検証を品質プロセスに組み込む
新技術の使い方や生産性の確認だけでなく、
安全に使えるものなのかという品質面を考慮して
データが増加すると 性能が劣化
エラーが発生したときの処理の仕方 など
事前検証をする。

ソフトウェア開発の達人の行動(1)



□ 達人は、技術的に難しい部分は人に任せない。

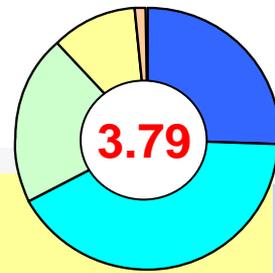
・ 難しいところほど、自分の仕事だと生き生きして設計する。

↔ **だめな人**: 難易度を考えないで、人に仕事を割り振る
難しいところを、他人に振る



- 難易度と人のスキルをチェックする
これこれ、あの人なら大丈夫という確認
- 設計書やプログラムには署名をする
署名することによって、責任感を持たせる
- 難しい部分を担当するのは、達人の生きがいでもある
全体を見てほしいからと、達人から生きがいである
難易度の高い部分の設計行為を取り上げるのは禁止

ソフトウェア開発の達人の行動(2)



- 達人は、標準化されたツールをうまく利用している。
- 達人は、楽をするための苦労は惜しまない。

- ・過去に作成した資産をうまく利用し、時間をかけなければいけない部分に十分に時間を使っている。
- ・必要であれば、ツールも作成して標準化する

↔ **だめな人:いつもイチからテンプレート作成。**

自分が作ったものでなければ気に入らない?

- ・ツールを利用すべき事項(静的解析ツールなど)は、ツールを利用する。
- ・時間がかかるツールは、昼休みなどに 実行させておく(時間の有効利用)

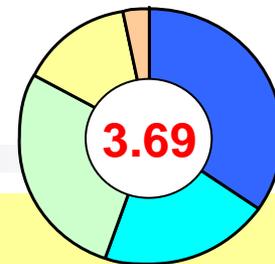
↔ **だめな人:いつも後々、問題が発生してから
静的解析ツールをかける**

- ツールは標準化し、どこに何があるかわかるようにする
- 静的解析などは、バッチ化して時間を有効活用する
- 必要なツールの利用を規則化し、最初から計画させる

標準化について(補足)

- 標準化の目的は「統一」と「単純化」であるが、つまりはそれによる「知識の再利用」である。すでに誰かが実施した「正しい・良い」ことを適用することである。
- それにより、“省思考”を推進し、“思考”リソースは新しいこと・難しいこと・重要なことにつぎ込むことができる。
- 単純にルールや決まりを目的にしてしまうのではなく、策定する側も 使う側も上記の視点でとらえることが重要である。

ソフトウェア開発の達人の行動(3)



□ 達人は、相手のスキルを見て設計する

・新人には・・・、初めて付き合う発注先には・・・、オフショアに発注する時には・・・と相手の実力を考えて設計内容を噛み砕いて記載している。

↔ **だめな人:次工程が誰かを考えないでいつも同じ設計あげく、思い通りにならないものが出来上がる。
「行間を読めないあいつらが悪い」**

●「次工程はお客様」である。

次工程はだれが、実施するのか？

次工程の人が、見てわかるのか？

思ったとおりの、ものができあがるのか？

という目で、設計レビューをする

相手のことを考えた行動について(補足)

- プロジェクトチーム内のことだけでなく、すべての作業や役割において、「次工程」さらには「その先の工程」を考えていくことは非常に重要。
(今さらわざわざ言うことではありませんが)
- しかし、現実にはそれができていないところも多いはず。

達人エッセンス利用による品質向上のまとめ



レビューでは、達人のエッセンスを利用しよう

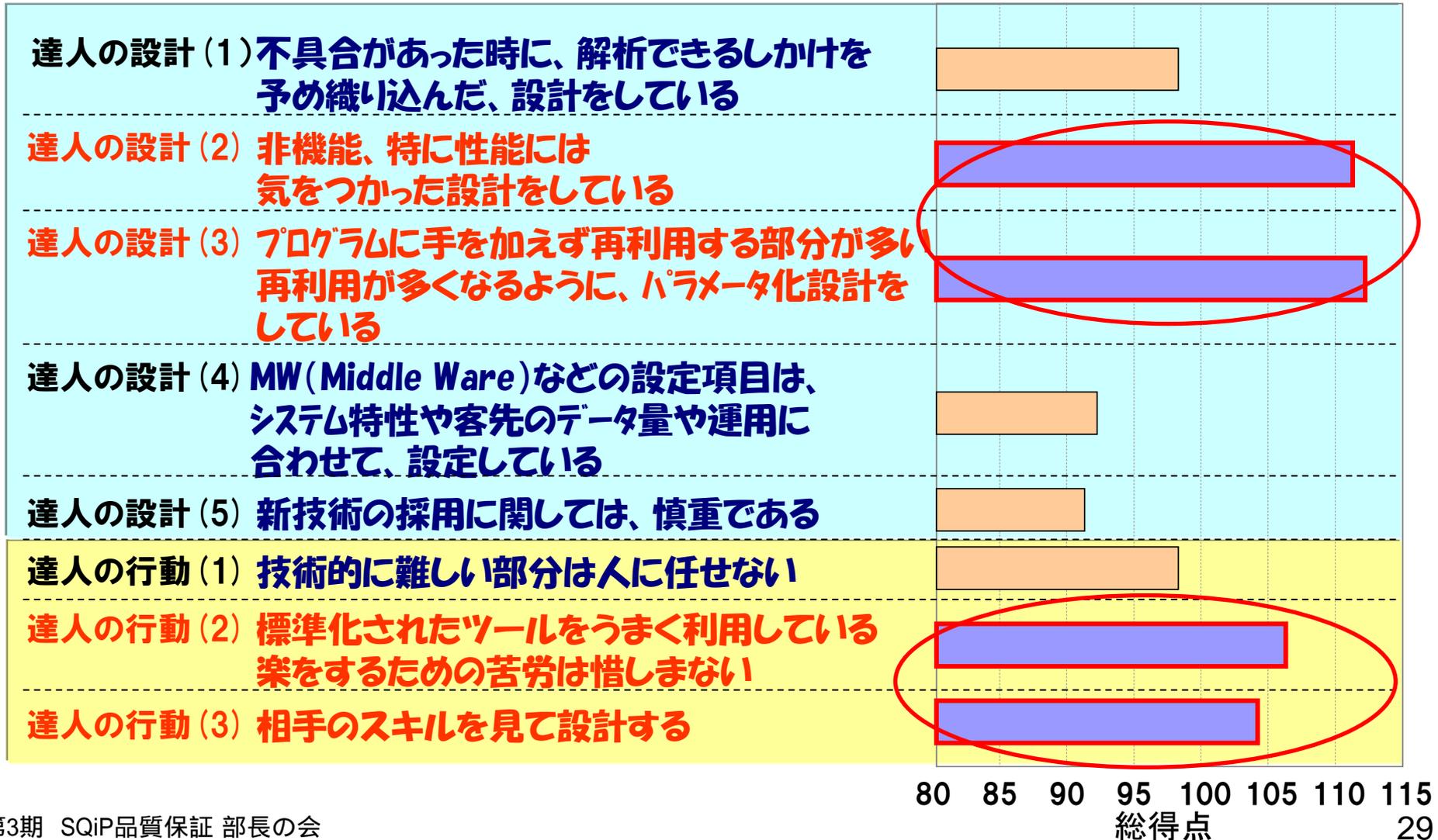
ログの仕掛け、性能定義と性能をクリアする設計、
客先運用条件、変更部分や共通設計の明確化 など

品質保証部門は、品質保証部門の達人になろう

- 達人が達人たる仕事をしてもらうために
設計環境の整備、ツールの標準化
なんでも達人にたのむのは禁止
- 発注先の実力や状態(忙しくて廻らないなど)を知っておく
- 各種データから、何が問題なのかを嗅ぎ分ける
- 分野別でだれが達人なのかを知っておき、困ったときに
レビュー参加依頼をする

アンケート結果(1)

全体的には、達人の設計(2)と(3)、達人の行動(2)と(3)が高い。



アンケート結果(2)

しかし、「部長の会」と「一般者」では相違点がみられる。
 「部長の会」は、達人の設計(4)と(5)の「重要度」が低い。
 「一般者」は、達人の設計(1)と(4)、行動(1)の「重要度」が低い。



「部長の会」と「一般者」の相違点の理由の想定

達人の設計(1) 部長の会:56% vs 一般者:31%

□ 達人は、不具合があった時に、解析できるしかけを
予め織り込んだ、設計をしている。

【一般者】 ログなどの仕掛けは十分にできていますよー。

【部長の会】 いやいや、システムが止まった時に、すぐに復旧させる
ログはいつも不十分だよ。

達人の設計(5) 部長の会:31% vs 一般者:69%

□ 達人は、新技術の採用に関しては、慎重である。

【部長の会】 採用する新技術は、そんなに多くはないと思うが？
さらに生産性をあげるため、新技術はどんどん採用
してほしい。

【一般者】 いやいや、まだまだ新技術はありますよ。それに
それで失敗すること、あるでしょ？

「部長の会」と「一般者」の相違点の理由の想定

達人の行動(1) 部長の会:69% vs 一般者:31%

□ 達人は、技術的に難しい部分は人に任せない。

【部長の会】 そうそう！ 当然、難しいところは、
ベテランの人がやるべきだよね。

【一般者】 いやいや、そんなことおっしゃられたって、
人はいないじゃないですか！

皆さんの会社ではいかがでしょうか？

一般者vs管理職

設計部門vs品質部門

重要だと思われる事項を話し合っ、改善してはいかが
でしょうか？

おわりに

すばらしい仕事をするには、
自分のやっていることを好きにならなくてははいけない。
スティーブ・ジョブズ

枠の中からどうやって飛び出すかが重要。
技術に感性を結びつけると、大きな飛躍ができる。
井深 大

日本のモノ創いは、古くから達人が支えてきた。

メイドインジャパンのソフトウェアの高品質化に向けて、もう1度、達人魂や「ココロ」の伝承に目を向けるべきではないでしょうか？

仕組みのみにこだわるのではなく、達人の能力をうまく活用するなどでの品質向上が必要でしょう。

品質保証部門は、その上でどのような対応をすべきかを良く考えましょう。



■Webアプリケーションの話

- J2EEデザインパターン …定番です

Java+Webのアプリケーションにおいて頻繁に使われる良い設計の
パターン集。

画面、ロジック、DBアクセス、メッセージングと一通りを網羅

- J2EE BluePrints

Designing Enterprise Applications with the J2EE Platform, Second Edition など
Java+Webアプリケーションの作り方のお作法です。

■システム全体の話

- ITアーキテクトのやってはいけない
- ITアーキテクトのやってはいけないvol.2

■こんな本もあります

- 達人プログラマー—システム開発の職人から名匠への道
- アンチパターン～ソフトウェア危篤患者の救出～
- デスマーチ～ソフトウェア開発プロジェクトはなぜ混乱するのか～
- ソフトウェア職人氣質—人を育て、システム開発を成功へと導くための重要キーワード