

ITプロジェクトに於ける品質管理の実態と課題

大和田 裕

品質について

品質とは

- 品物またはサービスが、使用目的を満たしているかどうかを決定するための評価の対象となる固有の性質・性能の全体。

(備考1)品物またはサービスが、使用目的を満たしているかどうかを判定する際に、その品物又はサービスが社会に及ぼす影響についても考慮する必要がある。

(備考2)品質は品質特性によって構成される。

(旧JIS Z 810D)

簡単に言うと 『品質とは使用者の満足度』

品質管理とは

- 買手の要求に合った品質の品物又はサービスを経済的に作り出すための手段の体系、品質管理を略してQCということがある。
- また、近代的な品質管理は、統計的な手段を採用しているので、特に統計的品質管理(Statistical Quality Control,略してSQC)ということがある。

(旧JIS Z 810D)

品質管理とは

- 品質管理を効果的に実施するためには、市場の調査、研究・開発、製品の企画、設計、生産の準備、購買・外注、製造、検査、販売及びアフターサービス並びに財務、人事、教育などの企業活動の全段階にわたり、経営者を始め管理者、監督者、作業員などの企業の全員の参加と努力が必要である。このようにして実施される品質管理は全社的品質管理(Company-Wide Quality Control,略してCWQC)又は総合的品質管理(Total Quality Control,略してTQC)という。
(JIS Z 810D)

品質保証とは

- 消費者の要求する品質が十分に満たされていることを保証するために、生産者が行う体系的活動。

(JIS Z 8101)

保証：確実に間違いないということを請け負い、責任をもつこと

補償：相手がこうむった損害などを補い償うこと

保障：生命・財産・権利などを保護して守ること

ソフトウェア品質

■ ソフトウェアにおいても、品質の考え方は同じ

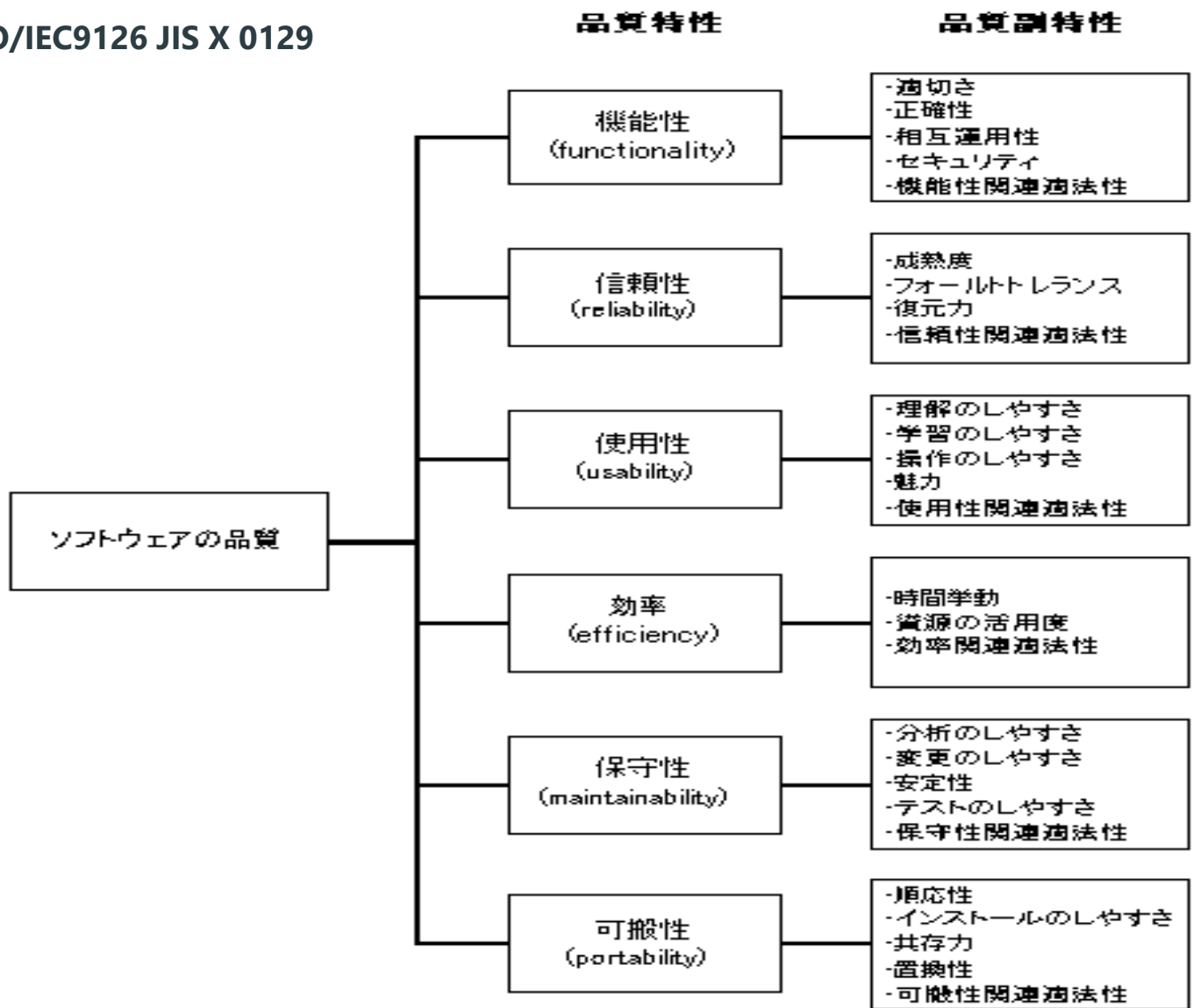
- QCDSME（狭義の品質・コスト・納期・安全性・モラル・環境）の総和
- 狭義の品質
 - 受入検査に象徴されるようにお客様が、製品・サービスを受け取ってくれるか否かの判断基準
- お客様受容の対策
 - 工程管理・進捗管理・技術者教育・営業資源の確保
- 対策と管理をすべて対象にして実施する企業活動全体が「品質管理」

ソフトウェアの特性

- 目に見えない
- 顧客のニーズ（ユーザ要求）が定義できない
- ソフトウェアの品質と評価がしにくい
- 個人依存度が高い
芸術 → 家内手工業 → ソフトウェア工業生産
- 技術のライフサイクルが短い
- ソフトウェアは特殊との意識が強い
- 業界として常に需要を抱えている
「ほどほど品質」の許容

ソフトウェア品質特性

ISO/IEC9126 JIS X 0129



ソフトウェア品質保証の変遷



品質管理の実際と課題

品質管理の問題

1. 品質管理は指標の中の1つ
2. 品質管理のばらばらな理解
3. 品質管理に対する誤解

品質管理は指標の中の1つ

- プロジェクトでの個々の問題は、それぞれがその時の優先順位を持っていて品質問題だけがいつも高い訳ではない。
- プロジェクト・マネージャは、プロジェクトの優先順位に従って処理する。
- プロジェクト・マネージャは、管理可能な変数(人、技術、時間、組織、ルールなど)を組み換えてプロジェクトを上手く進める様にしている。その結果を評価する指標の一つが品質がある。
- プロジェクト・マネージャにとって品質は主役ではない。主役は機能で、要求定義・設計の技術を駆使してシステムを作る。その評価観点として品質がある。

なぜ品質に焦点をあてるのか？

■ 経済性

- ✓ 多くのプロジェクトにおいて、計画性のない修正作業は最大コスト要因（全コストの40%-80%）
- ✓ 欠陥コストは欠陥がプロセス内に長く留まる程増大
 - 欠陥の作りこみ工程が上流であればあるほど、また欠陥の抽出が下流工程であればあるほど欠陥修正コストは増大する。
 - 要求定義工程で作られた欠陥を要求定義工程内で抽出した場合を1とすると、要求定義工程で作られた欠陥を出荷後修正すると50から200倍の費用が発生。

■ 品質はプロジェクトのなかに計画されている必要がある

- ✓ 最後に付け加えるのでは役に立たない。

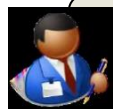
■ 変更は不安定さを増大させる、従って欠陥は早く除去する

品質管理のばらばらな理解

段階

成果

問題点



品質管理以前

事故・クレーム多発



検査重視

検査体制整備

テスト工程主体のQC

出荷品質向上
不良の摘出力向上
不良の早期摘出

工程遅延
開発効率低下



製造品質重視

作り込みの品質

不良の作り込み低減
開発効率向上

設計品質の低迷
費用対効果低減



設計品質重視

高信頼性技術

当たり前品質確保

ユーザ満足度不十分
魅力的品質不十分

高付加価値品質重視

魅力的品質達成
高収益達成

品質管理に対する誤解

1. 品質を向上するとコストアップする
2. 品質管理とは検査を厳重に行うことである
3. 品質管理とは標準化を行う事である
4. 品質管理とは統計的手法を用いてデータを解析することである
5. 品質管理とは品質管理部門がやることである
6. 品質管理は生産部門にやらせておけばよい
7. 品質管理とはQCサークルの事である

品質管理に対する誤解

1. 品質を向上するとコストアップ

■ 高い規格値の設定

- ✓ クリアーするための追加投資
- ✓ 不良の増加

■ 品質関係の専門家投入

- ✓ コストの追加

■ 管理業務の増加

- ✓ 間接コストの増加

■ 品質を向上させると言う事は、ばらつきを少なくする事

- ✓ 例えば、今まで基準からはずれるのが10%(5%+5%)あったものを、5%(2.5%+2.5%)以下にする。
- ✓ すなわち品質を向上してばらつきが少なくなればなるほど不良は減り、コストはさがる。

品質管理に対する誤解

2. 品質管理とは検査を厳重に行うこと

- 最終成果物の検査、中間成果物の検査を強化する
 - ✓ 不良は検査で取り除くという古い発想
- 不良の原因を取り除く事が重要
 - ✓ その原因を現場と不良を見ながら、考えて管理する事が品質管理

テスト工程の問題点

- 試験工程の大半が、試験そのものよりもデバッグに費やされている（補強工事の連続）
 - ✓ 試験は不良の摘出を重視しており、正しいことを保証しているとは限らない。
 - ✓ 試験工程の実体は、試験・デバッグ工程
 - ✓ もう一つの事実：不良修正のミスは20-30%

品質管理に対する誤解

3. 品質管理とは標準化を行う事

■ 標準化によって品質の基準を設定

- ✓ 基準の成果物をつくりだすための方法等を考え実践する仕組みを作り出すことが品質管理
- ✓ 最終成果物を標準化するためには、各工程の作業の管理点を標準化する必要がある。この管理項目が重要管理点。ハードルの高さ、ハードルの種類を明確にすることになる。
- ✓ この管理項目を定期的に確実に管理して、管理値に異常がでていないかを管理図などで管理

品質管理に対する誤解

4. 統計的手法を用いてデータを解析

- 統計的手法でデータを解析するのは、あくまで手段
 - ✓ 最終成果物の規格のばらつきを押さえるために、どの要因を押さえることが一番効いているか、次の要因は何が効いているかを押さえることが品質管理。

5. 品質管理とは品質管理部門がやること

- 品質管理部門は、データを出してその問題点を指摘するだけであって、造り込む行為・改善していく行為は、あくまでも現場が造り込むもの。

品質管理は生産部門が行う

6. 品質管理は生産部門が行う

- お客様受容の対策
 - ✓ 工程管理・進捗管理・技術者教育・営業資源の確保
- 対策と管理をすべて対象にして実施する企業活動全体が「品質管理」

品質管理とはQCサークル

7. 品質管理とはQCサークル

- 当社は品質管理を強化しますと発表し、○○○対策委員会を造りました。
- 現場では、QCサークルを行い全社で品質管理を強化します。そして、QCサークルを時間外で行い、発表会をして、品質管理を進めます。
- QCサークルの目的は現場の意識の向上であり、確かに教育活動の一環ではあるかもしれませんが、決して品質管理イコール、QCサークルではありません。
- 最終成果物の品質向上をどうしたらいいかを、現場単位のQCサークルに丸投げをしてしまうのです。

品質管理の難しさ

- 品質の管理というプロセスそのものが、常にその品質を管理されていなければならない
- 品質管理は常に変化するもの
 - 例えば、セキュリティの観点から常にソースコードをチェックしているチームは、全く新しい攻撃手法が公開されれば、これまで実行していたチェックのプロセスを改善する必要がある。

品質管理の難しさ

- **品質管理という作業のスコープがあいまい**
 - ✓ 品質管理に手をつけようとする、あらゆるところに問題を見いだすことが出来る。
- **品質の目標が分からない**
 - ✓ 品質を不良にかぎっても、コストと納期は定量的な目標値が決まっている。従って、予定・計画が立てられる。引き算で、現在の目標が分かる。
 - ✓ 一方、品質は定量的な目標値が難しい。品質（不良 摘出）は積み上げであり、不良摘出の目標値は設定しても、目標値を達成しても、品質の保証にはならない。
- **品質管理の費用対効果**
 - ✓ 製品は作れば売ることが出来るが、品質管理チームは製品の完成には寄与出来ても作り出すことは出来ない。

効果的な品質管理を目指して

効果的な適用

1. エンジニアリング（技術）
ソフトウェア工学に基づく正しい開発方法論により、品質を作りこむ
 2. マネジメント（管理）
品質管理の原点に戻り、PDCAを回し改善し続ける
 3. モラル（士気）
気力を充実させ、作業に集中できる職場環境を作る
- ✓ 正しい方針に基づく、充実した、やりがいのある、達成感を味わえる、チームワークのよい楽しい職場から、良いソフトウェアは生まれる。

- ファクトコントロール（定量的品質管理）
 - ✓ 定量的なデータに基づき正確な事実を把握する

定量的品質管理

- 科学的なものの見方と客観性に基づいた行動が必要
 - ✓ データを読み、データで語る。
 - ✓ 系統的・総合的・計画的に現象を把握し、分析・検討ができる。
 - ✓ KKD(経験と勘と度胸)は素晴らしいが、データによる裏付けが不可欠。裏付けがあってKKDは活きる
 - KKDだけだと、議論ばかりでまとまりがつかない
 - KKDだけだと、声の大きい人や役職が上位の人の意見がとおりやすい
 - KKDにたよると、試行錯誤に陥りやすい
 - ✓ 系統立てた考え方をするので、得られた結果から客観的な判断が下せる。
 - ✓ 統計的な裏付けがあるので、結果に説得力がある。
 - ✓ 客観的であるので意見が一つにまとまりやすい。
 - ✓ 日ごろ気付かない欠点が浮き彫りにされ、見落としがなくなる。

定量的品質管理は何のため

■ ソフトウェアの定量的品質管理は何のために実施するのか



- 上司やお客様に対する品質報告資料を作成するため。
- プロジェクト報告書に品質データを記載しなければならないから。
- PMや上長に指示されるし、社内ルールで決まっているから。



- できればやりたくない。その時間を開発活動に使いたい。
- 品質状況はデータなど無くても過去の経験や勘で分かるし。
- お客様への説得材料としては有効なので、やむを得ない。

形にこだわり過ぎると、

- レビュー時間や不具合数を計測し、基準値と比較することによって品質が確保できているかどうかを判定する活動です。

定量的品質管理の目的

- 定量データを使って問題を早期発見できれば、余裕を持って手が打てる。
- しかし、実際には多くのプロジェクトが定量データを上手く活用できていない。
- しかし、定量的品質管理というと



という4プロセスに注力しがちで、現場には分かりにくい。

肝心の「データを開発の途上で活用する」
ということも伝わらない

定量的品質管理の前提

- ソフトウェアにバグは必ず存在する。しかし、どのくらいあるかが分からない。
 - ✓ 人が作るものには必ずバグが混入する。
 - 問題は、どれくらい混入しているのか計測する手段がないこと。
 - ✓ そのため、「バグの検出が少ないから品質が良い」とは考えず、
 - むしろ「検出できずに残っているバグがあるのだ」と考えるべき。
 - ✓ プログラムのロジックとして正常でも、仕様に合わなければバグになる。
 - この種のバグは、設計段階で発見できなければ結合テスト以降まで残存しがち。

定量的品質管理の前提

■ ではどうするのか？

- ✓ バグが出なくなるまでテストする × 非効率すぎる
- ✓ 過去の定量データを使い、混入バグ数を推測する ○

一般的な混入量が分かれば、自分のソフトウェアの品質が良いのか悪いのか、どのくらいのバグを潰せば安心できそうか、といった見通しが立てられる。

定量的品質管理の前提

- ソフトウェアの定量的品質指標のほとんどは代替指標ではない。

- ✓ 言い換えれば、ソフトウェアの品質は直接計測できない、ということ。
- ✓ 品質を直接計測できないために代替指標を使うが、
 - レビュー時間が長いからといって、レビュー品質が良いとは限らない（レビュー密度）
 - バグを大量に検出して潰したからもう大丈夫だとは言えない（バグ密度）

といった課題がある。

代替指標では品質について断定や保証ができないため、

- 標準よりレビュー密度が低ければ、レビュー不足が懸念される
- バグ密度が標準より高ければ、何か問題があることが疑われる

品質そのものではなく、問題を早期に察知するための手段と考える。

定量的品質管理の前提（まとめ）

1. ソフトウェアの定量的品質指標は代替指標であり、その値で品質を評価するのではなく、品質問題が発生していないか開発の途上で察知するために活用する。
 - ✓ その方法には、トレンドを監視する・過去のデータと比較する・サブシステム間など同じプロジェクトで相互評価するなどがある。
2. 過去の同類プロジェクトの品質データを目安として活用することが重要。即ち、混入されている潜在バグ数の推測や、トレンドが異常な状態になっていないかどうかの評価に活用する。
3. 品質指標に少しでも異常があれば、その箇所を特定し、原因を掘り下げる。問題があれば対処するだけでなく、同様の問題が他に波及していないかも確かめる。
指標値の異常が常に品質問題に起因しているとは限らない。難易度の高いテストが初期段階に集中していれば、バグが最初から多く検出されることもある。

効果的に実施するためには

バグは作り込んだ工程で検出し、除去する

- 各工程で作り込んだバグは、原則としてその工程内で除去することを目指す。
 - ✓ 後工程になるほどバグの検出と除去にかかる手間やコストが増加し、改修を繰り返すことで新たな品質劣化を招いてしまう危険性が高まるため。
 - ✓ 特に上流工程（要件定義や基本設計）で混入されたバグは、下流に進むにつれて波及範囲が広がって除去するのが難しくなる。

効果的に実施するためには

- バグをその工程内で除去するには、混入しているバグの量を知る必要がある。
 - ✓ 過去の類似プロジェクトにおけるバグ密度が分かれば、対象となる成果物の規模を掛け合わせることで算出。
 - ✓ 実際には対象プロジェクトの特性を考慮する必要があるが、推測値に過ぎないので厳密に算出することに注力しすぎない。
 - ✓ 混入バグ数の推測値から工程毎のバグの残存数を管理し、工程審査などで活用。
- 完全除去は困難なので、残存バグをコントロールすること、前工程から引き継がれたバグが加わっていることを常に認識することが重要。

効果的に実施するためには

定量品質指標の監視は、その粒度を考慮する

- 品質データの異常を的確に把握するには、データを集約する粒度が重要。
 - ✓ 粒度が大きすぎると異常値が平準化されてしまい、細かすぎると監視しきれない。
 - ✓ また、プログラム、モジュール、サブシステムという階層だけでなく、機能別・チーム別など色々なカテゴリーで見ることが有効。

効果的に実施するためには

収集データの基準を統一しておく

- せっかくデータを収集しても、その基準がバラバラでは何も見えない。
 - ✓ きちんと傾向が見えなければモチベーションも下がり、計測が更にいい加減になるという悪循環に陥いる。
- 統一しておくべき項目には次のようなものがある。
 - ・ ステップ数のカウント方法
空行の扱い、セミコロン単位、言語の違いなど
 - ・ ドキュメント量
頁数／行数／バイト数など
 - ・ テストケース数のカウント方法
 - ・ レビュー時間
実施時間に人数分を掛けるか、工数にするか

効果的に実施するためには

定量データは開発途上で活用することに意味がある

- 定量的品質管理は第三者に対する品質報告のために実施するものではない。
 - ✓ 各種報告では、開発作業で活用した結果としての品質データを報告することに本当の意味がある。
 - ✓ 報告だけのために集めたデータでは説得力がない。
- テストが全て終了してから、「データ集計したらバグ密度が標準値を大幅に超えていた」というのでは手遅れ。
- せっかく集まったデータを、途上管理で活かすことが重要。

効果的に実施するためには

結果を現場にフィードバックする

- 収集されたデータで品質評価をした結果を現場にフィードバックする。
 - データを提出する側は、それがどう活用されているのかわからない。知らなければモチベーションも下がり、データの精度が落ち、提出期限も遅れ気味になる。
- 後続のプロジェクトのために品質データを蓄積する
 - ✓ 定量品質管理を上手く実施するには、過去のデータを活用することが不可欠。
- 今後、自分自身が別のプロジェクトで利用することも含め、最終的なデータは最後に精査し、組織として蓄積することが重要。

標準化のパラドックス

■ 標準化の意味

- ✓ プロジェクト遂行のために守るべき事項の共有・活用
- ✓ 関係者の誰が対応しても品質が許容範囲に収まる
- ✓ 欲しい成果を獲得するための媒体や手段として活用

■ 標準化で陥りやすい問題

- ✓ 標準化による思考停止
- ✓ 遵守すべきことが実践できていない
- ✓ 結果や成果にほとんど影響を与えない形式的対応

□ 対策

- ✓ “適合”ではなく“有効性”
- ✓ 自ら構築・展開し、効果確認&見直しを実施

ご清聴ありがとうございました