

## ソフトウェア品質の定義と変遷、 その測定の考え方

2009-6-25  
NARAコンサルティング  
奈良 隆正

1

## 本日お話しすること

- ソフトウェア品質の定義と変遷
  - そもそも測定すべき品質とは何かを今一度ふりかえります。
- ソフトウェア品質の測定とその留意点
  - 実際の測定にあたって、その概要と留意点をお話します。
  
- 目次
  - 【第1部】ソフトウェア品質の定義と変遷
  - 【第2部】ソフトウェア品質の測定とその留意点

2

## 背景-SQuBOKガイドから

- ソフトウェアの位置付けの変化
  - ITシステムは今や社会インフラの一部
  - ソフトウェアはすべての産業のコアテクノロジー
- ソフトウェア品質を起因とする事故の発生
  - ソフトウェア品質に起因するトラブルが後を立たない
  - ソフトウェアの位置付けの変化に伴い、経済活動への影響が拡大
- 日本のソフトウェア品質向上への取り組み不十分
  - ソフトウェア品質向上への取り組みは、各社に依存
  - ソフトウェア品質向上の技術(方法論)は、未整理
  - 先人の知識や経験は、暗黙値のまま
  - ソフトウェア品質向上技術は将来最も重要な技術

3

## ソフトウェア品質の基本概念とマネジメント

(出展:SQuBOKガイド)

- 品質の概念
  - 品質の定義(先人の洞察)
  - ソフトウェア品質モデル(歴史、標準化)
  - 更にシステムとしての品質へ  
ディベンダビリティ、セキュリティ、ユーザビリティ、セーフティ
- 品質のマネジメント
  - 品質のよい製品・サービスを提供するために組織を指揮し、管理すること
  - 品質コントロール、品質保証、改善、測定・評価
  - ソフトウェアの品質マネジメントの特徴
  - V&V (Verification & Validation)、検査

4

## SQuBOK®に垣間見る日本の品質管理の特徴(1/2)

- 「品質」
  - 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」
- 「品質保証」
  - お客様に安心して使って頂けるような製品を提供するための全ての活動
  - ※プロダクトとプロセスが、特定された要求に合致しているかどうかという観点の十分な確信を提供する活動ではない
- 「改善」
  - 不十分でもとにかく動き出して全員が今より高いところを目指してプロセスそのものを改善しながら進める
  - ※欧米流の、プロセスを定義してその通りに実行していることを確認することではない
- 「品質第一」
  - 品質を高めることによって手戻りや作業のムダを省き、継続的な納期の短縮やコストダウンにつなげていく
  - ※納期を厳守するために必要な作業を省くのではない

5

## SQuBOK®に垣間見る日本の品質管理の特徴(2/2)

- 「品質の作り込み」
  - より上流で“悪さ”の知識を子細に活用し障害を排除していく
  - ※ただ単にきちんと作る、という意味ではない
- 「次工程はお客様」
  - 他の工程を助けるような改善を進め、全体最適へと向かっていく
- 「人間性尊重」「組織活性化」「小集団活動」
- 「5ゲン主義」
  - 現場・現物・現実 + 原理・原則
- 「全員参加」
  - 品質管理はスタッフだけではなく、現場も経営陣も一丸となって進めなければならない
  - 関係者全員が納得してベクトルをあわせ、目標達成のために取り組む
  - 現場中心のため隔々まで改善が行き届くとともに、全員が自ら考えて行動する組織を構築できる

6

## ソフトウェア品質知識体系ガイドとは？

### ■ SQuBOK®ガイド

Guide to the Software Quality Body of Knowledge



日本発のBOK!

SQuBOK策定部会 編  
2007年11月 オーム社より出版  
3500円

cf. PMBOK®ガイド：プロジェクトマネジメント知識体系ガイド  
(A Guide to the Project Management Body of Knowledge)

SWEBOK® : ソフトウェアエンジニアリング基礎知識体系  
(Guide to the Software Engineering Body of Knowledge)

7

## 【第1部】

## ソフトウェア品質の定義と変遷

測定の話に入る前に、そもそも測定すべき品質とは何でしょうか？  
ソフトウェアの品質、品質保証について、これまでに経験した知識が、どの  
ように定義、表現されてきたのかを振り返ります。

#注

【KA(n)やS-KA (n.n.n) の表示が有るスライドはSQuBOKガイドからの引用です】

8

## 品質の定義の変遷(1/3)

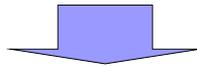
1. JIS(JIS)Z8101 品質管理用語)およびISO8402では、品質を以下のように定義している。

### ●JISの品質の定義:

『品物又はサービスが、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。』

### ●ISOの品質の定義:

『製品またはサービスが明示または暗黙のニーズを満たす能力として有している特徴および特性の全体。』



「品質とは、ユーザの要求(ニーズ、使用目的)を満足させるために製品(含むサービス)のもつべき特性である。」

9

## 品質の定義の変遷(2/3)

### ■ クロスビー(P. Crosby)の定義:

- 『品質は「要求に対する適合」である。』

### ■ ワインバーグ(G.M.Weinberg)の定義:

- 『品質は誰かにとっての価値である。』

ソフトウェア工学の  
専門家による定義

### ■ 品質保証という観点からの品質

- 品質にはユーザにとっての「価値」である。

### ■ 近代的品質管理における品質

- 品質は、ユーザにとっての「満足度」である。
  - 「満足度(CS:カスタマー、サティスファクション)」は、アメリカのマルコム・ボルトリッジ国家品質賞のスローガンであり、日本に逆輸入された

10

## 品質の定義の変遷(3/3)

### ■ 英国PRAXIS社(ソフト企業)のマーチン・トーマスのコメント

- 『カスタマー、サティスファクション(顧客満足度)は、我々のゴールである、しかしカスタマー(顧客)の視点は毎月でも変わりうる。』

### ■ ジェームス・マーチンの定義

- システムが本稼動する時、どこまで真のビジネス(ユーザ)ニーズに合っているかということ。
  - 開発期間が長年にわたるソフトウェア開発を意識した考え方、完成時にはユーザニーズ自体が変化しているとの認識  
⇒RADの必要性

11

## 品質の概念—KA(1.1)

### ■ (1)利害関係者の視点による定義(Garvin)

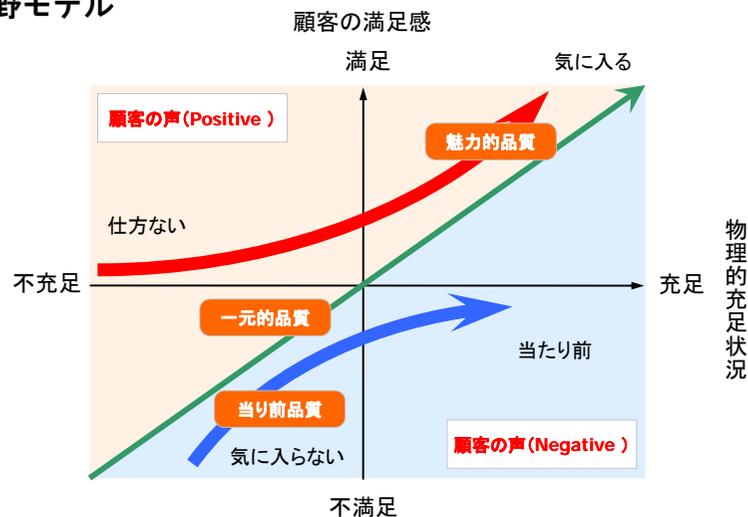
- 超越した視点:品質が何かを認識はできるが定義は困難である
- ユーザの視点:品質が目的に適合しているか
- 製造者の視点:品質が仕様に準拠しているか
- 製品の視点:品質が固有の製品特性に結び付いているか
- 価値に基づいた視点:品質は顧客が対価として支払う額に依存する

### ■ (2)満足感と物理的充足度合いによる定義(狩野)

- 当たり前品質要素:
  - それが充足されれば当たり前と受け取られるが、不十分であれば不満を引き起こす品質要素
- 一元的品質要素:
  - それが充足されれば満足、不十分であれば不満を引き起こす品質要素
- 魅力的品質要素:
  - それが充足されれば満足を与えるが、不十分であってもしかたがないと受け取られる品質要素

12

## 狩野モデル



出展：MOTテキスト製作委員会

## 魅力的品質

それが充足されていれば満足を与えるが、不充足であってもしかたがないと受けとられる品質要素。

### 一元的品質

それが充足されれば満足、不充足であれば不満を引き起こす品質要素。

### 当り前品質

それが充足されれば当たり前と受けとられるが、不充足であれば不満を引き起こす品質要素。

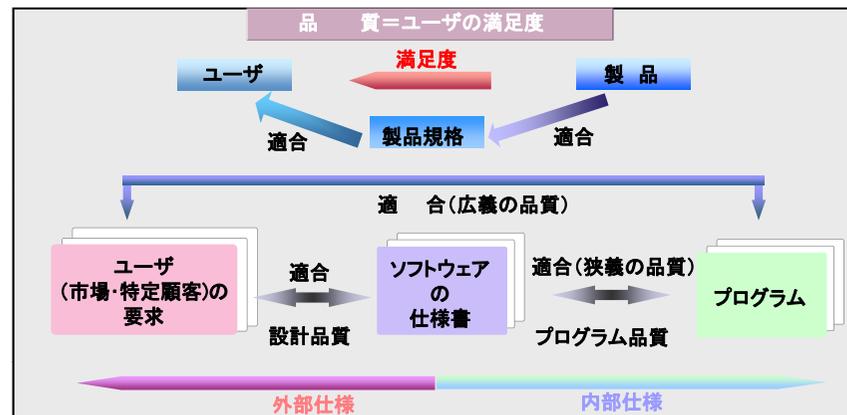
## 品質の概念－KA(1.1)

### ■ そのほか、色々な人の色々な定義

- ● Joseph M.Juran: 二つの視点
  - プロダクトの特性 (Feature) が顧客のニーズに応えることで満足を提供する
  - 不備: deficiencies (障害や誤り) から免れる
- ● Roger S.Pressman: ソフトウェアの品質を記述するのは困難といいつつ
  - 機能および性能に関する明示的な要求事項、明確に文書化された開発標準、および職業的に開発が行われた全てのソフトウェアに期待される暗黙の特性に対する適合
- ● Robert L.Glass: 品質は製品によって変わり全てに共通の品質の定義はない
  - 各製品が備えるべき一式の品質特性のことであり、プロジェクトの種類に応じて異なった優先度がつくべきもの
- ● 石川馨: 「狭義の品質」と「広義の品質」⇒ Quality を「質」と捕らえる
  - 狭義の品質: 製品の品質 ⇒ 欧米流の考え
  - 広義の品質: 仕事、サービス、情報、工程、部門、人、システム、会社の全てを含めた質

## ソフトウェア品質の考え方(広義と狭義)

- ソフトウェアを中心とするシステム開発において品質を考える際は、**狭義と広義**の両面から考える必要がある。**狭義**には、「ソフトウェア仕様書に記述(定義)された機能が実現されている事の確認」であり、**広義**には「ユーザ要求への適合度、すなわちシステム完成度の確認」である。



## 品質の概念－ KA (1.1)

- ソフトウェア品質を議論する場合の要点
  - 「時間」についての概念
  - 「コスト」の概念
  - 「必要な機能」が実現していること
  - 「使いやすい」こと
  - 「想定されている時間内」に機能を実行できること
  - 「実行時に故障を起こさない」こと
  - 「導入や学習」が容易なこと
  - 「導入後の維持や拡張」が容易なこと

17

## 品質の概念－ KA (1.1)

- ISO,IECなどにおける品質の定義
  - ISO9000:2005  
本来備わっている特性の集まりが、要求事項を満たす程度
  - ISO/IEC 25000:2005  
指定された特定の条件で利用する場合の、明示的又は暗示的なニーズを満たすソフトウェア製品の能力
  - ISO/IEC9126-1:2001、14598-1:1998  
ある“もの”の明示された、または暗黙の必要性を満たす能力に関する特性の全体(ISO8042:1994用語の定義を使用)
  - IEEE Std 610-12:1900
    - (1)システム、コンポーネント、またはプロセスが指定された要求を満たしている度合い
    - (2)システム、コンポーネント、またはプロセスが顧客またはユーザのニーズ(必要性)または期待を満たしている度合い

18

## 品質保証の定義(私の経験①)

- 消費者の要求する品質が十分に満たされている事を保証するために、生産者が行なう体系的活動。  
(JISハンドブック 品質管理 1995)
- ある“もの”が品質要求事項を満たすことについての十分な信頼感を供するために、品質システムの中で実施され、必要に応じて実証される、すべての計画的かつ体系的な活動。  
(ISO8042品質一用語 1994)
- 品目又は製品が、定められた技術的な要求事項に適合することにより、十分な信頼を得るために必要な、すべての計画的体系的な活動の型。  
(ANSI/IEEE729 ソフトウェア工学用語集 1994)

19

## 日本的な品質保証の定義(私の経験②)

- 品質保証は品質管理の真髄。消費者が安心して、満足して買うことができ、それを使用して安心感、満足感をもち、しかも長く使用することができるという品質を保証すること。  
(石川 馨 1981)
- お客様に安心して使っていただけるような製品を提供するためのすべての活動。  
(飯塚先生 2005)
- 品質保証活動とは、品質リスク(バグがあるかもしれない)を最小にする活動である。バグが無いことを保証する活動ではない。  
(西 先生 2006)

20

## 私なりの品質保証の定義(私の経験③)

### ■ これらの定義を私なりに解釈すると...

1. ユーザ(消費者)が満足する製品またはサービスの品質を保証するための組織的、体系的活動。
2. 品質保証はアクティビティ(活動)でありワーク(作業)ではない。管理のPDCAが廻らないと上手く行かない。
3. ソフトウェア品質保証は:  
全工程、全組織・全員参加、多岐に渡る活動

### □ <<良くある誤解>>

- **品質保証は品質保証部門の仕事であり、そこが頑張るもの。**

21

## 品質保証の考え方(①)－ S-KA (1.2.2)

### ■ 「品質保証」:

用語の意味は国や地域によって必ずしも同一ではない

#### □ ISO9000:2005の定義:

- 品質要求事項が満たされるという確信を与えることに焦点を合わせた品質マネジメントの一部
- 品質マネジメント:品質に関して組織を指揮し、管理するために調整された活動。

⇒品質計画、品質管理、品質保証、品質改善から構成

- 品質保証の活動範囲は品質マネジメントのうち、品質計画、品質管理、品質改善を除いた部分

22

## 品質保証の考え方(②)－ S-KA (1.2.2)

### ■ 「日本の品質保証」:お客様を満足させる活動の総称

- お客様に安心して使っていただけるような製品を提供するための全ての活動
- 品質保証は品質管理の真髄
- 活動内容について「実証」することを必ずしも重要視していない
- お客様が満足したという結果をもって品質保証活動の成果を測る

### ■ 「欧米的品質保証」:

- 品質保証していることの「実証」重要視して発展  
(契約社会という文化的な背景がある)

- 欧米の品質保証の解釈と日本の品質保証に対する解釈には大きな差がある
- 但し、顧客満足を目的とした活動と位置付ける点では同じである

23

## 改善の考え方－ S-KA (1.2.4)

### ■ 「改善」:ボトムアップ的かつ系統的現場活動

- 仕事の改善、プロセスの改善、製品品質改善がある
- 改善計画と目標を立て目標に向けた改善計画を実行
- 改善の成果がどう実現されたか評価し、再度計画を立て、これを繰り返す  
⇒PDCAや改善(KAIZEN)の方法として確立された
- 品質改善は現状の品質を把握(測定)し、ビジネス目標に合致する品質目標が重要

### ■ PDCA (T-1.2.3.1)

- 計画(Plan→目標、目的)、実施(Do→計画に基き)、評価(Check→結果調査)、改善(Action→評価結果に基き処置) プロセスを順に実施し、最後の改善を次の計画に結びつける
  - 【目的】:全体のレベルアップを図る
  - 【効果】:目的を合理的、効果的に達成する
  - 【留意点】:管理項目と目標値の決定、データ採取、データに基づく論理的判断、作業標準の見直し、再発防止の処置などが重要

24

## ソフトウェアの特徴

### ■ ソフトウェアの特徴

項番	特徴と問題点	信頼性向上への配慮
1	<b>&lt;論理の集合であること&gt;</b> ・論理の正確な設計が困難 ・論理の信頼性の高いテストが困難 ・正確な開発規模の見積りが困難	(1) 構造設計とそれに基づくレビュー (2) システムマッチングなテスト
2	<b>&lt;目に見えないこと&gt;</b> ・品質管理が困難 ・工程管理が困難	(1) 品質のビジュアル化 (2) 開発工程のビジュアル化
3	<b>&lt;個人への依存が高いこと&gt;</b> ・個人差が大きい ・多人数の共同作業	(1) 開発手法の標準化と自動化 (2) 再利用技術 (3) 教育
4	<b>&lt;ユーザーズと直結していること&gt;</b> ・ユーザーズの正確な理解が困難 ・使用条件の正確な把握が困難 ・なかなか仕様が決まらない ・システムは生き物である(成長する)	(1) 要求仕様分析手法 (2) 実使用条件でのテストによる検証 (ex. System Simulation Test)

25

## (ご参考)ソフトウェア工学の問題

### ■ ソフトウェアには:

- 複雑性がある
  - ソフトウェア開発の難しさの元
- 順応性がある
  - 如何なる課題にも対応できる
- 不可視性である
  - 目に見えない
- 変更可能性がある
  - 仕様が決らなくても先に進める

26

## ソフトウェアの品質マネジメントの特徴(1) - SKA (1.2.4)

### ■ ハードウェアの品質管理技法をそのまま適用することが難しい

- ソフトウェアが物理的実態を持たない
- ソフトウェアは設計の自由度が高く論理矛盾等障害を作り込みやすい
- ソフトウェアは測定すべき物理特性が殆ど存在しない
  - そもそも、何を測定すべきかについて統一した見解がない
- ソフトウェアの仕様は殆どが自然言語で記述される
  - 曖昧性が高く矛盾を引き起こしやすい
- ソフトウェア開発は人間の知的作業によって行われる
  - モチベーションが大きく作用し品質、生産性に影響を与える
- ソフトウェア開発はチーム開発である
  - チームワーク、リーダーシップ、コミュニケーションが重視される

27

## ソフトウェアの品質マネジメントの特徴(2) - S-KA (1.2.4)

### ■ 規模の増大が二つの問題を生む

- それぞれの技術者が全体を把握できない
  - 開発の見通しが悪くなることで障害を作り込みやすい
- 開発人数が増大しコミュニケーションの密度が低下する
  - チームワークの確保が難しい

知的作業の質に悪影響を及ぼし、信頼性を低下させる

### ■ ソフトウェアエンジニアリングの充実が必要

- 設計の自由度を高めないようにする
  - 定石やデザインパターン等の活用
- 品質向上に寄与する指標の検討
- 仕様のレビューや管理の充実
- 開発の“悪さ”の知識を抽出し、体系化して蓄積

28

## 【第2部】

# ソフトウェア品質の測定と、 その留意点

ソフトウェア品質の測定と結果の活用は高品質なソフトウェアの開発に必要な不可欠です。しかしながら、目的が明確ではない測定や、測定条件が異なる他の組織の基準値の安易な利用は、必ずしも高品質なソフトウェア開発にはつながってはいきません。

第2部では、この“ソフトウェア測定”と“メトリックス”に焦点をあて、本来どのような測定し、活用すべきなのかを過去の経験とSQuBOKからお話します。

#注

【KA(N)やS-KA(NNN)で始まるスライドはSQuBOKガイドからの引用です】

29

## 第2回プロジェクト実態調査

(出展：日経コンピュータ、2008、12月1日号)

### □ 定量的管理と成功率

— 有り：45.6%

— 無し：24.3%

### □ 定量的管理の実態 (%)

(2008 2003)

成果物 18.0 8.7 51.9 46.4

コスト 14.8 5.1 63.2 76.2

進捗 29.9 12.8 54.6 54.9

### □ 遵守率 (%)

(2008 2003)

□ プロジェクト成功率 : 31.1% (26.7%、2003)

30

## はじめに

### ■ 製品開発における測定はコアコンピタンス

- 「物、事を計測し評価する能力」
- NASAが使う部品の信頼度はsix nine⇒計測できたことが凄い

### ■ 計測を確実にするために

- メトリックスが必要
  - ソフトウェアメトリックスは代用特性である
  - メトリックスは必要性和目的に合わせて定義されなければ成らない
  - メトリックスは常に見直しが不可欠である
- 計測結果は活用されなければ意味がない
  - 活用の基本はプロセスへのフィードバックである
  - 活用されないデータは精度が劣化してゆく

### ■ 最近散見される問題

- 慣習によって意味を理解せず計測していないか？

31

## ソフトウェア測定の考え方とメトリックス(1/2)

### ■ 計測は品質管理、プロジェクト管理のみならず、あらゆる管理(制御)の基本である(私の経験)

- 「測れないものは制御できない(トム・デマルコ)」

### ■ 計測を確実にするために

- 計測が容易であること
  - 計測の容易(可能)な尺度を選択
  - 計測を容易にする仕掛けとツールの導入
- 計測実績データの蓄積、評価
  - 計測条件(コンテキスト)の明確な実績データの収集、蓄積
    - 計測条件が異なるデータ(絶対値)の比較は意味が無い
  - 実績データの分析、評価
  - 評価基準の設定

32

## ソフトウェア測定の考え方とメトリクス(2/2)

- **メトリクスは代用特性であり、常に整合性を評価することが重要。メトリクスは品質特性およびプロセスとの結びつきが明確であること(先人の経験)**
- **経験的、品質特性のメトリクス**
  - 信頼性メトリクスの例を次のスライドに掲示
- **経験的、プロセスのメトリクス**
  - プロセスメトリクスの例を次のスライドに掲示

33

## 品質特性(信頼性品質指標)メトリクスの例

品質特性	副特性	計測尺度	定義
成熟性 無欠陥性	誤り密度	誤り密度	$\frac{\text{誤り件数 [件]}}{\text{成果物の量 [行]}}$
		残存誤り密度	$\frac{\text{最終成果物に含まれた誤り件数 [件]}}{\text{最終成果物の量 [行]}}$
	試験密度	試験密度	$\frac{\text{テストの量 [項目]}}{\text{成果物の量 [行]}}$
		エラー収束率	$\frac{\text{総摘出エラー数}}{\text{信頼度成長曲線の飽和点 (＝推定総エラー数)}}$ [個]
	信頼性	試験実施率	$\frac{\text{実施されたテストの量 [項目]}}{\text{予定されたテストの量 [項目]}}$
		ダウン発生率	$\frac{\text{ダウンに至った回数 [回]}}{\text{発生障害件数 [件]}}$
	誤り許容性	誤入力誤操作検出率	$\frac{\text{システムのチェック機能によって検出された誤入力・誤操作の数}}{\text{記録された誤入力・誤操作の数}}$ [回]
		平均故障発生間隔	$\frac{\text{システムの総稼働時間 [時間]}}{\text{観測された故障発生件数 [回]}}$
	可用性	稼働率	$\frac{\text{稼働状態にあった総時間数 [時間]}}{\text{観測時間数 [時間]}}$
		平均ダウン時間	$\frac{\text{稼働状態にあった総時間数 [時間]}}{\text{観測されたダウンの回数 [回]}}$
平均エラー修正時間 (注1)		$\frac{\text{エラーの修正に要した時間の総和 [人時]}}{\text{観測中に修正されたエラーの総件数 [件]}}$	
平均復旧時間		$\frac{\text{復旧に要した総時間 [時間]}}{\text{観測されたダウンの回数 [回]}}$	

(注1) 平均エラー-寿命ともいう。

34

## プロセスメトリクスの例(1/3)

項番	工程	品質指標	単位	管理レベル	備考
1	基本設計	1. ドキュメント不良件数	件	プログラム単位	
		2. ドキュメント不良件数/頁	件/頁	プログラム単位	
		3. DR指摘件数/頁	件/頁	プログラム単位	
		4. 投入工数/頁	人H/頁	プログラム単位	
		5. 検完遅延日数	日	プログラム単位	
		6. ドキュメント検査実施回数	回	プログラム単位	
2	機能設計	1. ドキュメント不良件数	件	プログラム単位	
		2. ドキュメント不良件数/頁	件/頁	プログラム単位	
		3. DR指摘件数/頁	件/頁	プログラム単位	
		4. 投入工数/頁	人H/頁	プログラム単位	
		5. 検完遅延日数	日	プログラム単位	
		6. 変更指示件数	件	プログラム単位	
		7. ドキュメント検査実施回数	回	プログラム単位	
3	構造設計 / 詳細設計	1. ドキュメント不良件数	件	プログラム単位	
		2. ドキュメント不良件数/頁	件/頁	プログラム単位	
		3. DR指摘件数/頁	件/頁	プログラム単位	
		4. 投入工数/頁	人H/頁	プログラム単位	
		5. 検完遅延日数	日	プログラム単位	
		6. テスト項目数/KS	個/KS	プログラム、モジュール単位	
		7. 変更指示件数	件	プログラム単位	
		8. ドキュメント検査実施	回	プログラム単位	

35

## プロセスメトリクスの例(2/3)

項番	工程	品質指標	単位	管理レベル	備考
4	コーディング	1. 投入工数/ステップ	人H/ステップ	プログラム単位	
		2. フラグ数	個	プログラム、モジュール単位	
5	単体テスト	1. 不良摘出件数(累積)	件	プログラム、モジュール単位	成長曲線
		2. 不良摘出件数/KS	件/KS	モジュール単位	
		3. テスト項目進捗度	%	モジュール単位	
		4. 不良摘出件数/投入工数	件/人H	モジュール単位	
		5. 不良摘出件数/マシン時間	%	モジュール単位	
		6. テストプログラム消化率	%	モジュール単位	
		7. 不良率(不良摘出件数/テスト項目数)	%	モジュール単位	

36

## プロセスメトリックスの例 (3/3)

項番	工程	品質指標	単位	管理レベル	備考
6	プログラムテスト	1. 不良摘出件数(累積)	件	プログラム、モジュール単位	成長曲線  (デバッグ効率)
		2. 不良摘出件数/KS	件/KS	モジュール単位	
		3. テスト項目進捗度	%	モジュール単位	
		4. 不良摘出件数/投入工数	件/人H	モジュール単位	
		5. 不良摘出件数/マシン時間	%	モジュール単位	
		6. テストプログラム消化率	%	モジュール単位	
		7. 設計総合耐久テストの故障率	件/H	システム、プログラム単位	
		8. 探針での不良率	%	プログラム単位	
		9. 探針での故障率	件/H	プログラム単位	
		10. 探針での推定残不良数	件	プログラム単位	
		11. 不良率(不良摘出件数/テスト項目数)	%		
7	製品検査	1. 不合格件数	件	システム、プログラム単位	(設計発見不良)
		2. 不合格件数/KS	件/KS	プログラム単位	
		3. 製品検査実施回数	回	プログラム単位	
		4. 故障率(1/MTBF)	件/H	プログラム単位	
		5. MTBD	H/件	プログラム単位	
		6. 不良率	%	プログラム単位	
		7. 不良摘出件数(累積)	件	プログラム単位	

37

## 古典的な3大メトリックス

- ◆ 開発規模
- ◆ 開発工数
- ◆ バグ数

- 未だに取り続けているが有効に活用されているか疑問
- 未だに取っていない組織がある、驚き！
- SEI(CMMIチーム)はコアメトリックスと呼んでいる  
～ 新しいことのように言っている ～

38

## ソフトウェア品質特性 (ISO/IEC 9126シリーズ) ①



図1.1.1 外部及び内部品質のための品質モデル

[出典: JIS X 0129-1:2003 ソフトウェア製品の品質 - 第1部: 品質モデル, p. 9 図4]

39

## 品質特性の概要 (1/2)

### 1. 機能性 (functionality)

機能の集合の存在及びそれらの明示された性質の存在をもたらす属性の集合  
機能は、明示的又は暗示的な必要性を満たすものとする

### 2. 信頼性 (reliability)

明示された条件の下で、明示された期間、ソフトウェアの達成のレベルを維持するソフトウェアの能力をもたらす属性の集合

### 3. 使用性 (usability)

- 明示的又は暗示的な利用者の集合が、使用するために必要とする労力及び個々の使用結果による評価に影響する属性の集合

40

## 品質特性の概要（2 / 2）

### 4. 効率性 (efficiency)

明示的な条件の下で、ソフトウェアの達成のレベルと使用する資源の量との間の関係に影響する属性の集合

### 5. 保守性 (maintainability)

仕様化された改定を行うために必要な労力に影響する属性の集合

### 6. 保守性 (portability)

ソフトウェアのある環境から他の環境へ移す際の、そのソフトウェアの能力をもたらす属性の集合

41

## 【参考】ソフトウェア品質特性（ISO/IEC 9126シリーズ）②

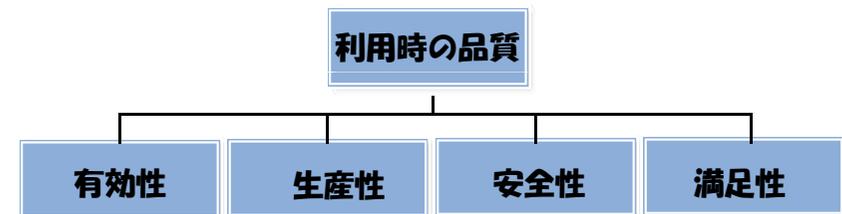


図 1.1.2 利用時の品質のための品質モデル

[出典：JIS X 0129-1:2003 ソフトウェア製品の品質  
- 第1部：品質モデル, p. 13図5]

42

## ソフトウェア品質特性（メトリクス例）-1/6

### 1. 機能性 (functionality)

機能の集合の存在及びそれらの明示された性質の存在をもたらす属性の集合  
機能は、明示的又は暗示的な必要性を満たすものとする

- 合目的性；ユーザ改良件数、仕様の改定率
- 正確性；ユーザから要求された数値精度の実現率、  
説明書と実動作の合致度
- 相互運用性；データ形式の合致度、インターフェースや  
プロトコルの合致度
- セキュリティ；暗号化率、アクセス履歴保有率
- 機能性標準適合性；関連する全ての規格中、遵守している  
項目の割合

43

## ソフトウェア品質特性（メトリクス例）-2/6

### 2. 信頼性 (reliability)

明示された条件の下で、明示された期間、ソフトウェアの達成のレベルを維持するソフトウェアの能力をもたらす属性の集合

- 成熟性；平均故障間隔 (MTBF)、平均故障寿命 (MTTF)
- 障害許容性；誤入力、誤操作検出力、単位障害あたりの  
ダウン回数
- 回復性；平均修復時間 (MTTR)、平均ダウンタイム (MDT)
- 信頼性標準適合性；関連する全ての規格中、遵守している  
項目の割合

44

## ソフトウェア品質特性(メトリクス例)-3/6

### 3. 使用性 (usability)

- 一 明示的又は暗示的な利用者の集合が、使用するために必要とする労力及び個々の使用結果による評価に影響する属性の集合

- 理解性；デモが用意されている機能の数
- 習得性；マニュアル装備率、学習機能装備率
- 運用性；メッセージ的確度、キータッチ数、レスポンス時間
- 魅力性；利用者にとって魅力的である機能の数
- 信頼性標準適合性；関連する全ての規格中、遵守している項目の割合

45

## ソフトウェア品質特性(メトリクス例)-4/6

### 4. 効率性 (efficiency)

- 一 明示的な条件の下で、ソフトウェアの達成のレベルと使用する資源の量との間の関係に影響する属性の集合

- 時間効率性；処理速度（CPU実行時間、入出力処理時間など）、処理能力（トランザクション処理件数など）
- 資源効率性；資源使用量（CPU使用量、メモリ使用量など）、資源使用率（単位時間当たりの、CPU利用率、メモリ利用率など）
- 効率性標準適合性；関連する全ての規格中、遵守している項目の割合

46

## ソフトウェア品質特性(メトリクス例)-5/6

### 5. 保守性 (maintainability)

- 一 仕様化された改定を行うために必要な労力に影響する属性の集合

- 解析性；誤り箇所識別率、診断機能の装備率
- 変更性；KLPCあたりの修正実施時間、障害1件あたりの修正実施時間
- 安定性；モジュールの結合度
- 試験性；KLPCあたりのテスト時間、障害1件あたりのテスト時間
- 保守性標準適合性；関連する全ての規格中、遵守している項目の割合

47

## ソフトウェア品質特性(メトリクス例)-6/6

### 6. 保守性 (portability)

- 一 ソフトウェアのある環境から他の環境へ移す際の、そのソフトウェアの能力をもたらす属性の集合

- 環境適応性；適用可能機種／OS数、変更せずに使用できるデータの割合
- 設置性；ソースコードの変更率、移行ツールの適用できる割合
- 共存性；共通資源を他のソフトウェアと共存できる割合
- 置換性；置換前のソフトウェアと比べて、実現できる機能や性能の割合
- 保守性標準適合性；関連する全ての規格中、遵守している項目の割合

48

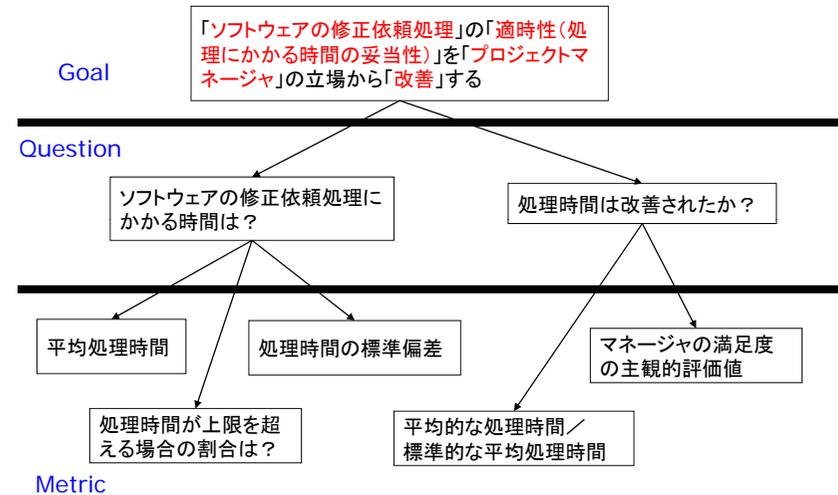
## 【参考】メトリクス定義の代表的な技法

### ■ GQMパラダイム

- Basili教授らによって提案された総合的なソフトウェア計測の枠組み
- “計測はトップダウンに行われるべきである”
  - まず計測目標があって、その目標を遂行するために尺度(メトリクス)が定義され、計測されるべき
- “データ分析は何らかの目的や仮説に基づいて行われるべきである”
  - 例)コスト予測, コスト改善, 品質改善 etc.

49

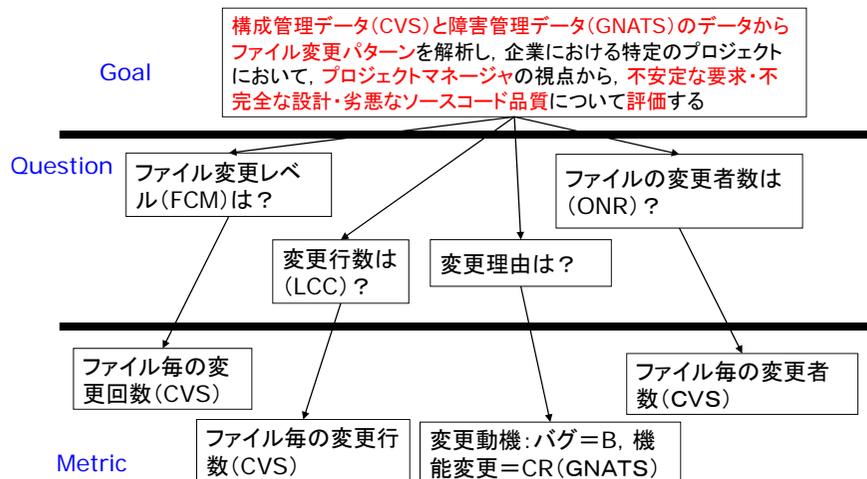
## GQMモデル例\*



\*井上克郎, 松本健一, 飯田元 著「ソフトウェアプロセス」共立出版, 2000.

50

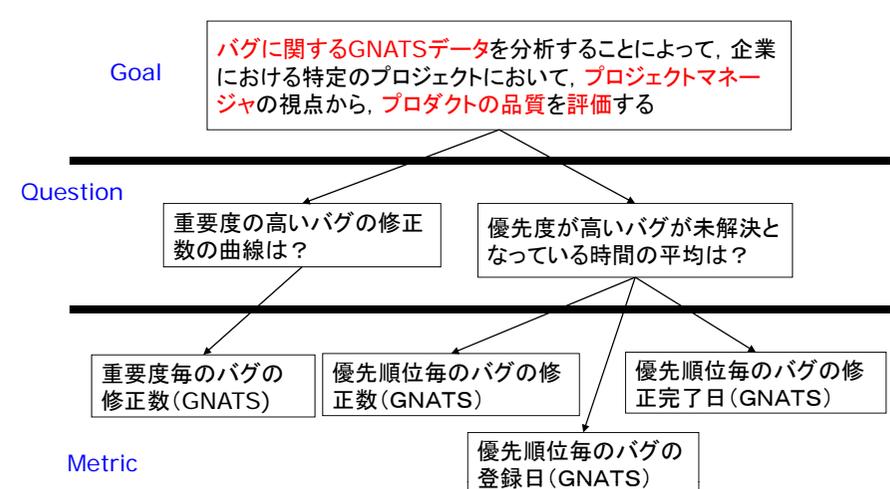
## GQMモデル(1)



出展: EASEプロジェクト

51

## GQMモデル(2)



出展: EASEプロジェクト

52

## メトリックス－KA (3.1)

“もの”の測定可能な特徴を属性と呼び、属性を測定する方法と尺度を合わせた概念の集合がメトリックス

### ■ 2種類のメトリックス

- プロダクトメトリックス;製品の属性を測定する
  - ソフトウェア品質メトリックス、規模メトリックスなど
- プロセスメトリックス;プロセスの属性を測定する
  - 開発プロセスの個々の作業や手順全体に関するメトリックス
  - 作業に影響を与える人や組織といった開発基盤に関するメトリックス

### ■ メトリックスを用いる目的;

- 製品やプロセスの品質を定量的に把握、評価し、継続的に改善すること
- 製品とプロセスの両方に焦点を当て、改善することが重要
- どのようなメトリックスを用いるかは、何を目的にどのような目標を設定するかで決定

53

## ソフトウェア測定の実態－ S-KA (1.2.5)

- ソフトウェアの測定はISO/IEC 15939:2002で定義
- メトリックスには基本メトリックスと導出メトリックスがある (JIS X 0141では基本測定量と導出測定量)
- 測定にあたり;
  - 目的、方法(操作の場合)、尺度、活用プロセスを明らかにする
  - 測定目的とメトリックスを結びつけることが重要(例:GQM)
  - 属人性を排した形で測定方法を定義することが重要
  - 名義尺度、順序尺度、間隔尺度、比率尺度のどれを用いるか定義
- 測定プロセスの共通的な枠組み(ISO/IEC 15939:2002);
  - 測定に対するコミットメントの確立および保持
  - 測定プロセスの計画
  - 測定プロセスの遂行
  - 測定の評価

● 測定プロセスとは、測定の目的や方法、尺度の定義、選択、適用を改善するために必要な手順

54

## 測定理論 ～用語の定義(1)～ S-KA (3.1.1)

### ■ 測定尺度;

- 測定可能な特徴を示す属性を計量するもの
  - 一定の尺度で物事を測定することにより、客観的な評価や判断を可能とする

### ■ 品質メトリックス;

- ソフトウェアの品質特性、副特性を定量的に計測するもの
  - ソフトウェア製品の品質特性、副特性を定義し、測定、評価を可能とする

### ■ 測定値;

- 属性に割り当てられた「値」(直接測定値と間接測定値)
  - 測定可能な特徴を値で示し、測定対象の実体や状況を具体的に把握する

### ■ 指標;

- 予測や見積り、評価等の情報ニーズに基づいて示す数値または変数
  - 予測や見積りの際の目安や補助、組織やプロジェクトの達成度合いの評価基準に用いる

### ■ 評定水準;

- 測定尺度を分類するために使われる順序尺度上の点
  - ソフトウェア種別や利用者の要求等を考慮して、評価目的ごとの品質要求水準を明確にする

55

## 測定理論 ～用語の定義(2)～ S-KA (3.1.1)

### ■ 評価基準;

- プロセス、製品、プロジェクト、または資源の価値や品質を評価するときの基準
  - 品質特性または品質副特性の評定水準に基づいて、ソフトウェア製品品質の総合評価が客観的に行えるようにする

### ■ 測定プロセス;

- メトリックスを実際に計測するプロセス
  - 開発済みのソフトウェアメトリックスを計測し、当該ソフトウェア自体の評価、他ソフトウェアとの比較、今後の保守のスケジュール、投入人月の見積りに使用する。
  - 開発中のソフトウェアメトリックスを計測し、開発プロセスを動的に制御しながら、ソフトウェアを計画どおりに効率よく開発する「ソフトウェア開発の計器飛行」を可能にする

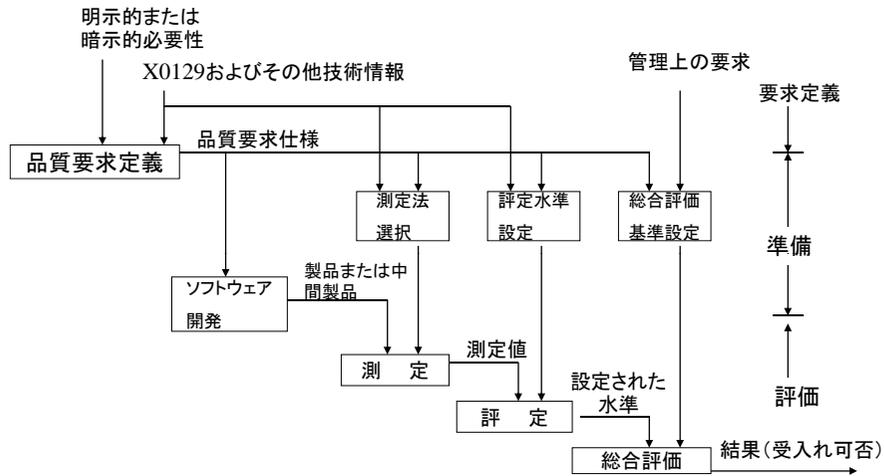
### ■ 評価プロセス;

- ソフトウェアメトリックスを使用して計測した結果を評価する
  - ソフトウェア製品の品質評価を計画的に実施して、客観的、効率的な評価が行えるようにする

● 参考規格:ISO/IEC15939:2002、9126-1:2001、9126-2:2003、  
TR9126-4:2004、14598-1:1998、2382-14:1997、  
IEEE std610.12-1990、IEC60050-191:1990

56

## ISO/IEC9126 品質評価プロセスモデル



出典)東基衛(他編):『ソフトウェア品質評価ガイドブック』、日本規格協会、1994年。

57

## 測定理論 ～留意事項(1)～ S-KA(3.1.1)

### ■ 測定尺度:

- 独自に定義した測定尺度を用いる場合には、其の定義や測定手順等を明らかにし、測定尺度を共有する関係者間で十分な認識合わせを行う必要がある

### ■ 品質メトリクス:

- (1)分解された品質の副特性を単一の値で表現することは難しく、複数の測定値から総合的に評価する必要がある
- (2)複数の測定値に重みづけして、単一の代表値を算出すると、計測した値の本来の意味が失われる危険性がある
- (3)品質の副特性と測定値の相関関係や精度をトレースする必要がある
- (4)品質の副特性への分解には、いくつかの方法があり、したがって、計測値も、一意には定まらない
- (5)測定を負担のかかり過ぎない計測方法を採用すべきである
- (6)例えば、ユーザビリティのように、定量的な測定が困難な品質特性もある。その場合は、主観的評価や、聞き取り調査等の方法も考慮する

58

## 測定理論 ～留意事項(2)～ S-KA(3.1.1)

### ■ 測定値:

- 測定値(データ)と使用した測定尺度や属性は常にセットで示し、“データの一人歩き”を避ける配慮が必要である

### ■ 指標:

- 指標と一緒に、その確信度や重要性等を定量的に示し、分析、評価の手助けとする

### ■ 評価水準:

- 品質は与えられた必要性によって決定されるものであるため、評価水準は品質の評価目的に応じて設定する必要がある

59

## 測定理論 ～留意事項(3)～ S-KA(3.1.1)

### ■ 評価基準:

- 評価基準の確立と運営には、組織的な評価技術の管理が不可欠である

### ■ 測定プロセス:

- どの時点で、どの要素を、どのように計測するかを予め明確にする

### ■ 評価プロセス:

- メトリクスによる測定は場当たりに実施するのではなく、組織やプロジェクトごとに明確に定義されたプロセスの中へ組み入れて実施すべきである

(補足)測定プロセスはISO/IEC 15939に基づいてソフトウェア一般の測定、評価プロセスを、評価プロセスはISO/IEC 14598-1に基づいてソフトウェア製品品質の測定、評価プロセスを解説している

60

## まとめ① 品質の定義と品質保証

### ■ 1. 品質の定義と解釈

- ① 品質は長い間、「要求に対する適合」とされてきたが、近年は「ユーザにとっての価値」、「ユーザに取っての満足度」、「真のビジネスニーズに合っている度合い」などが主流である。
- ② 日本は、仕事、サービス、情報、工程、部門、人、システム、会社の全てを含めた質を表す傾向がある。

### ■ 2. 品質保証—この用語の解釈は、国や地域で異なる。

- ① 日本的品質保証：お客様に安心感を与え、お客様を満足させるための全ての活動の総評であり、必ずしも活動内容を実証することを重要視していない。
- ② 欧米的品質保証：品質保証活動をしていることの実証を重要視している。これは契約社会という文化的な背景が影響している。

61

## まとめ② ソフトウェア測定

### ■ ソフトウェア測定

- ① 計測は品質管理、プロジェクト管理のみならず、あらゆる管理(制御)基本である。
- ② 計測を確実にするためには、計測が容易であること、計測実績データの蓄積、評価が重要である。
- ③ 測定にあたり、目的、方法、尺度、活用のプロセスを明らかにすることが重要である。

**計測事象 = 因果関係 + バイアス + 偶発性**

(出展:長崎大中村先生—JASPIC講演より)

62

## まとめ③ ソフトウェアメトリクス

### ■ メトリクス

- ① メトリクスは、属性を測定する「方法と尺度」を合わせた概念の集合と考えられる。
- ② メトリクスを用いる目的は、製品やプロセスの品質を定量的に把握・評価し、「継続的に改善をする」ことである。
- ③ メトリクスによる評価は、単一ではなく複数を用いて総合的に行うことが望ましい。
- ④ 測定値は測定条件(コンテキスト)が異なると意味が変わってくる。  
測定値は方法、尺度、属性をセットで示し”データの一人歩き”を避ける配慮が必要である。

63

## 終わりに

- (1) ただ何となく慣習に従って計測してませんか
- (2) 何故このメトリクスが必要か考えたことはありますか
- (3) 計測結果はフィードバック(活用)されていますか  
⇒ 先ず、データを採取したプロジェクトとメンバーへ  
⇒ そして、プロセスへの反映
- (4) 数値目標をクリアすることが目的になってませんか  
(本来の施策が実施されていますか?)  
～メトリクスと測定を考え直してみよう～

～ご清聴有難う御座いました～

64