ソフトウェアの品質保証の本質
～　技法の変遷から学ぶ　～

２０１７年2月２２日
NARAコンサルティング
奈良　隆正

今日お話すること

- はじめに

1. ソフトウェア（SW）品質保証の定義
2. 70年代のSW品質保証活動・方法
3. 80年代のSW品質保証活動・技法
4. 90年代前半のSW品質保証活動・技法
5. 90年代中半のSW品質保証活動・技法
6. 2000年代のSW品質保証活動・技法

今日お話すること

- はじめに

1. ソフトウェア（SW）品質保証の定義
2. 70年代のSW品質保証法
3. 80年代のSW品質保証法
4. 90年代前半のSW
5. 90年代中半のSW
6. 2000年代のSW品

黎明期

隆盛期

定着＆成長期(前半)

停滞期(中半～後半)

再生期⇒発展期？

はじめに

品質保証で思い浮かべることは？

開発者を自称する皆さんからこんなネガティブな声が聞えそう

- 無理難題を押し付ける品質保証部門。
- 膨大なテストそしてまたテスト。
- どの様に使われるか良く解らない数多くのデータ取得
- チェックリストを埋めるためのチェック
- 何か特別（余計）な追加作業が必要
- 品証部門を立ち上げた、これで品質は大丈夫
- 品証の指摘はごもっとも、でも解決策はどうするの？
- 品質関連の数字は良くなっているらしいが、何か変わったの？
- 品証部門の本来のタスクが良く見えません。

皆さんの組織の現状は？

1. QMS（標準）は有りますか？
   （1）全社もしくは事業部単位に有る。
   （2）部門単位で有る。
   （3）上記のいずれも有る。
   （4）もしかすると無い。

2. QMS活用の実態は？
   （1）ほぼその通りに行われている。
   （2）参考にはしている。
   （3）ほぼ使われて居ない（飾ってある）。

1.1品質保証の定義(1/2)(ISO9000:2005)

- 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ージメントの一部

  - 品質マネージメント: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ージメントのうち、品質計画、　品質管理、品質改善を除いた部分

＊＊＊詳細は“SQuBOKガイド”参照＊＊＊

1.1品質保証の定義(2/2)

- 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品質ー用語　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）

1.2日本の品質保証の定義①

- 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。
  （西　先生　2006）

1.2日本の品質保証の定義②

これらの定義を私なりに解釈すると:

- 製品(サービス)を使って貰うユーザに「安心感」を持って頂くために開発者、提供組織が行うべき諸活動
- 開発者の仕事は全て品質保証に直結している
- 品質保証はアクティビティ(活動)である。ワーク(作業)ではない。
- ソフトウェア品質保証は;

  全プロセス、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、そこが頑張ってやるもの。

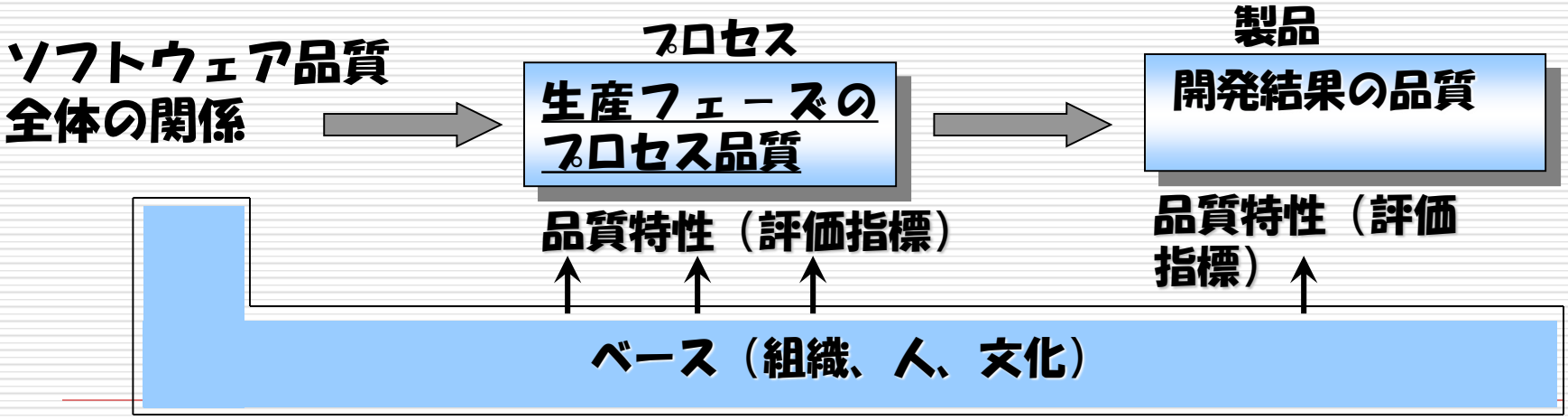
【補足】ソフトウェア品質保証の捉え方

ソフトウェア品質

製品品質
・開発結果のソフトウェアが製品として使用目的にどの程度あっているか
・ソフトウェア自身が使用目的を満たす為にどのような性質をどの程度持っているか

プロセス品質
・ソフトウェアを開発する生産プロセスの品質

開発基盤の品質
・人、組織、開発技術、管理技術、開発環境等の品質



●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

- 「日本的品質保証」;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」を重要視して発展（契約社会という文化的な背景がある）

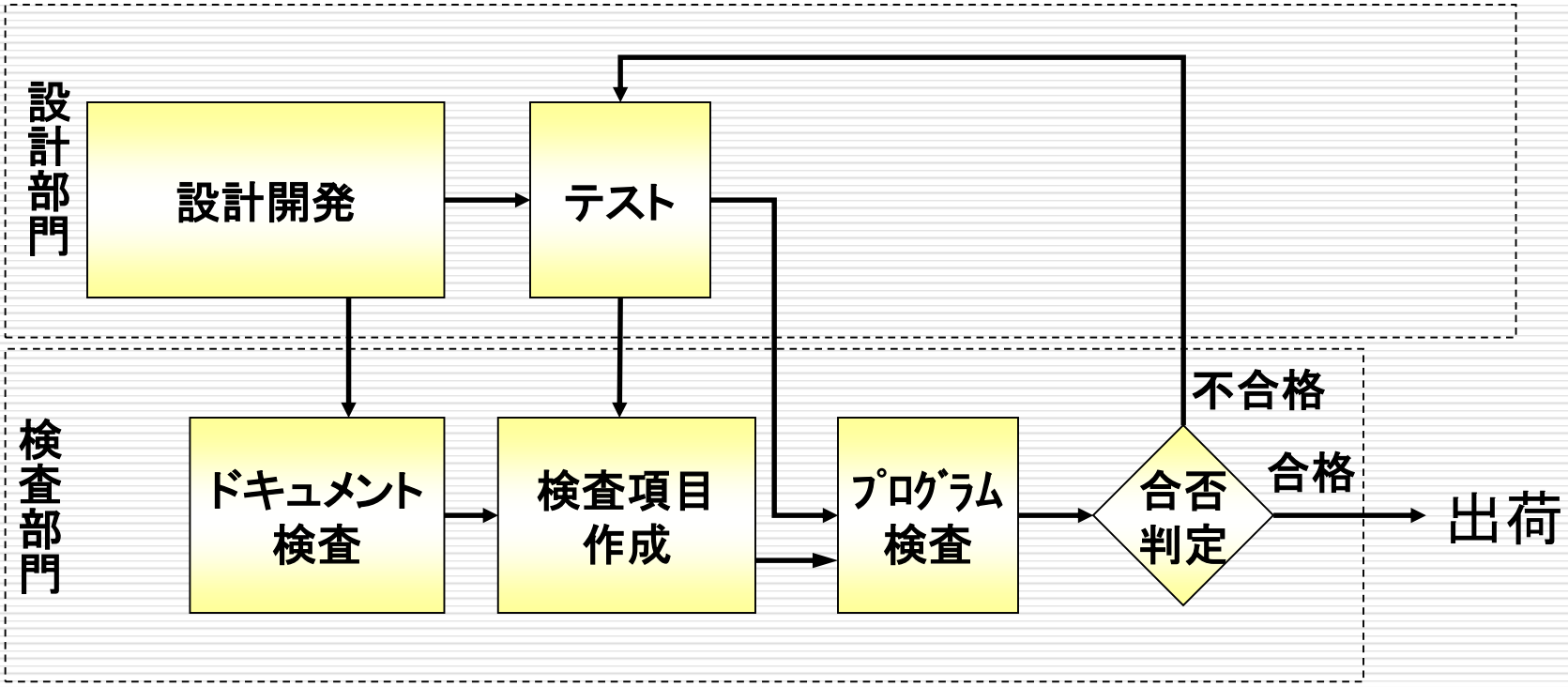
【参考】ソフトウェア工学の問題

- 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

2.1 1970年代の品質保証活動

- ハードウェア品質管理手法の導入　←ソフトウェア品質管理未熟
  - ソフトウェア工場（ファクトリー）制度導入
    標準化されたプロセス（手順）を繰り返すことで組織に
    開発のノウハウを蓄積し製品の品質を向上させていく手法。
  - ソフトウェア検査部門の独立化（確立期）
    最終成果物の検査：①不良品を出荷しない水際作戦
    （開発とQAの仁義無き戦い、体力勝負）
    ⇒ソフトウェアの素性を明らかにしたドキュメントが必要
  - 中間成果物のドキュメント検査：最終品質の予測
    ⇒ドキュメント検査は、V&Vの「検証」に相当

【補足】ソフトウェア工場制度における検査



開発工程と検査工程の関係

3．1 1980年代品質保証活動①

- 背景：ソフトウェア開発量の爆発
  → 水際作戦（完成品検査重視）から開発フェーズの品質管理重視へ
- ソフトウェア工場（ファクトリー）制度の定着、隆盛期
  - **製品完成度重視**
  - **日本的品質管理（PDCA、QC７つ道具、全員参加、小集団活）の導入**
- ソフトウェア検査部門の独立（発展、定着期）
  - **検査部門は開発の全プロセスに対して関与**

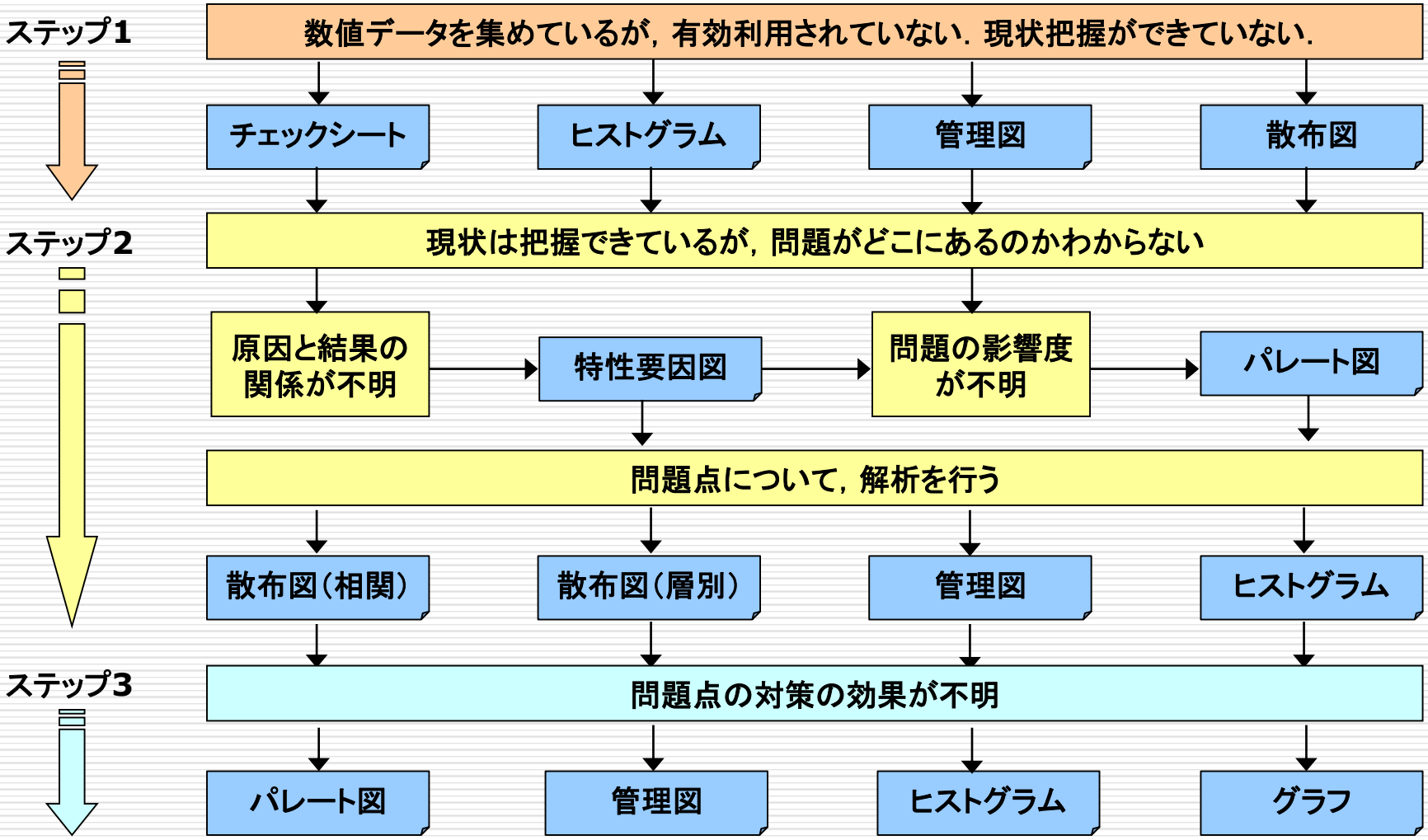
☆☆☆クスマノ（MIT）は「日本のソフトウェア戦略」で
日本の高品質ソフトウェアの源と評価　☆☆☆

(1) QC7つ道具

#	項目	内容
1	**特性要因図**	品質の特性と要因の関係を表す。普通、魚の骨に類似した形状となる。
2	**パレート図**	項目を横軸とし、度数の多い項目から順に度数を縦軸にとり、累積相対度数曲線を併用した図。
3	**チェックシート**	データの分類項目別分布を知るために使う。要因の系統的整理に際して、効率よくデータを取るのに有効。
4	**ヒストグラム**	データのばらつきの姿を知る。
5	**散布図**	2変数を横軸と縦軸にとり、値を打点して作られる図。相関性など、二つの対になったデータの検討に使う。
6	**管理図**	工程が安全な状態にあるかどうかを調べるため、または工程を安定な状態に保持するために持ちる図。
7	**グラフ**	データを数字ではなく図に表して、見やすくするために使われる。

表１　ＱＣ７つ道具

（2）QC七つ道具の使い分け



QC７つ道具の使い分け

（3）小集団活動

小集団活動とは：

「職場第一線のメンバーの、小集団をベースとした、
全員参加・自主管理方式による、職場改善活動」

つまり、職場第一線のメンバーが小さなグループを組み、
PDCAを回しながら職場や業務の改善をする活動。

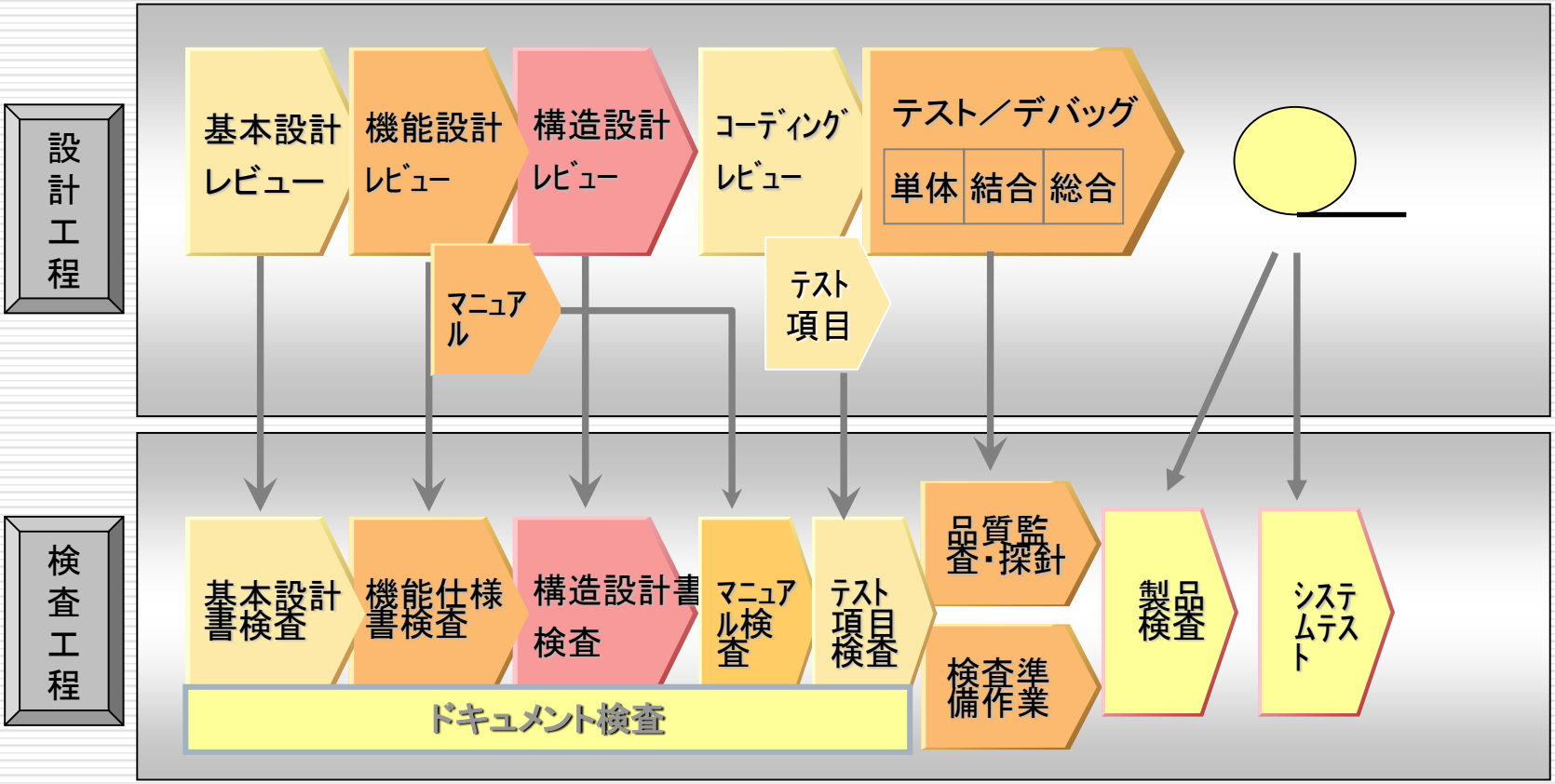
この活動を企業・組織で行なうことで・・・

1. 変化に対応できる試行性・機動性・弾力性のある職場づくり
2. 仕事の中に創造性・挑戦性を発揮する新しい職場づくり
3. 仕事と能力開発の結合を目指す新しいOJTの場づくり

出展：(株)ブレーン・ダイナミックス
http://www.brain-d.co.jp/

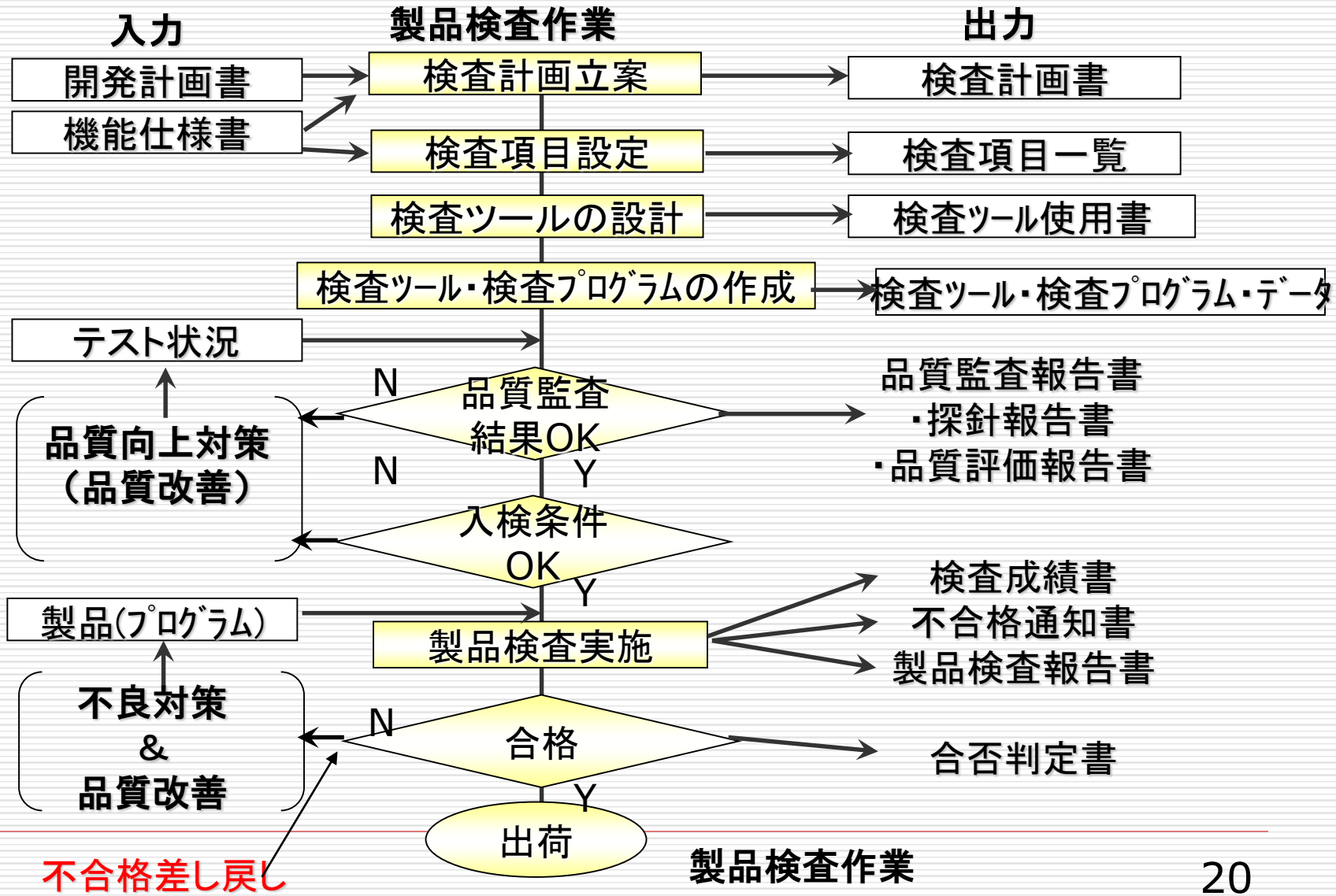
（4） 開発工程とテスト／検査の関係

（1）ウォーターホール型の開発



設計工程と検査工程の関係

【補足】ソフトウェア製品検査（80年代に確立）



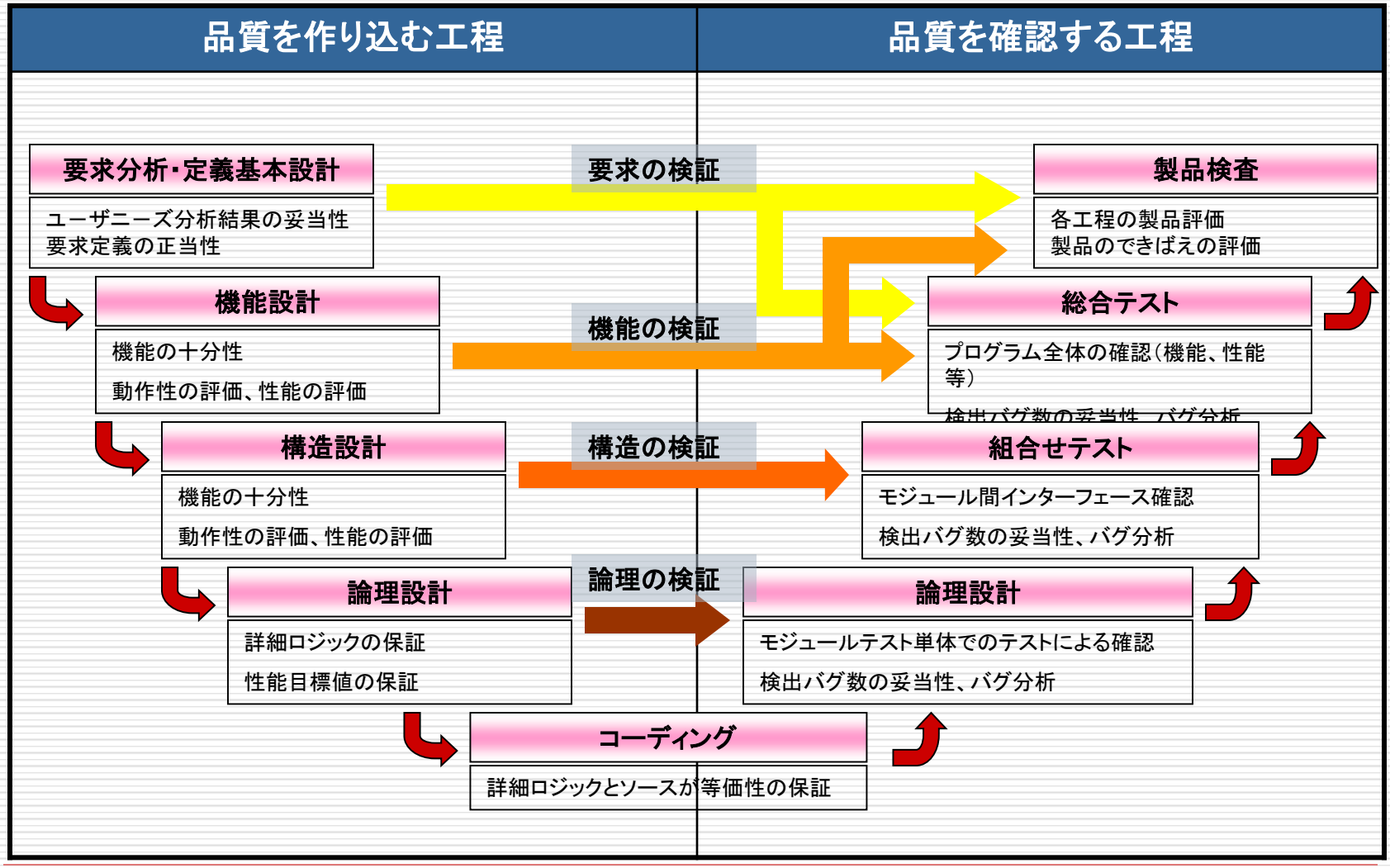
3.2 1980年代品質保証活動②

- ソフトウェア品質管理の体系化と普及
  - V字モデルに基づく管理体系の構築、普及
  - メトリックスや管理技法の開発、普及
    - 基幹3大メトリックス（バグ、規模、工数）
    - バグ数、テストの目標値の採用
    - 品質目標値管理手法
    - SRGM，抜取り検査法（探針）
    - バグ分析と水平展開
    - バグ発生/解決曲線、テストケース消化曲線の採用（品質管理図）

＜＜メトリックスの必要性と重要性＞＞

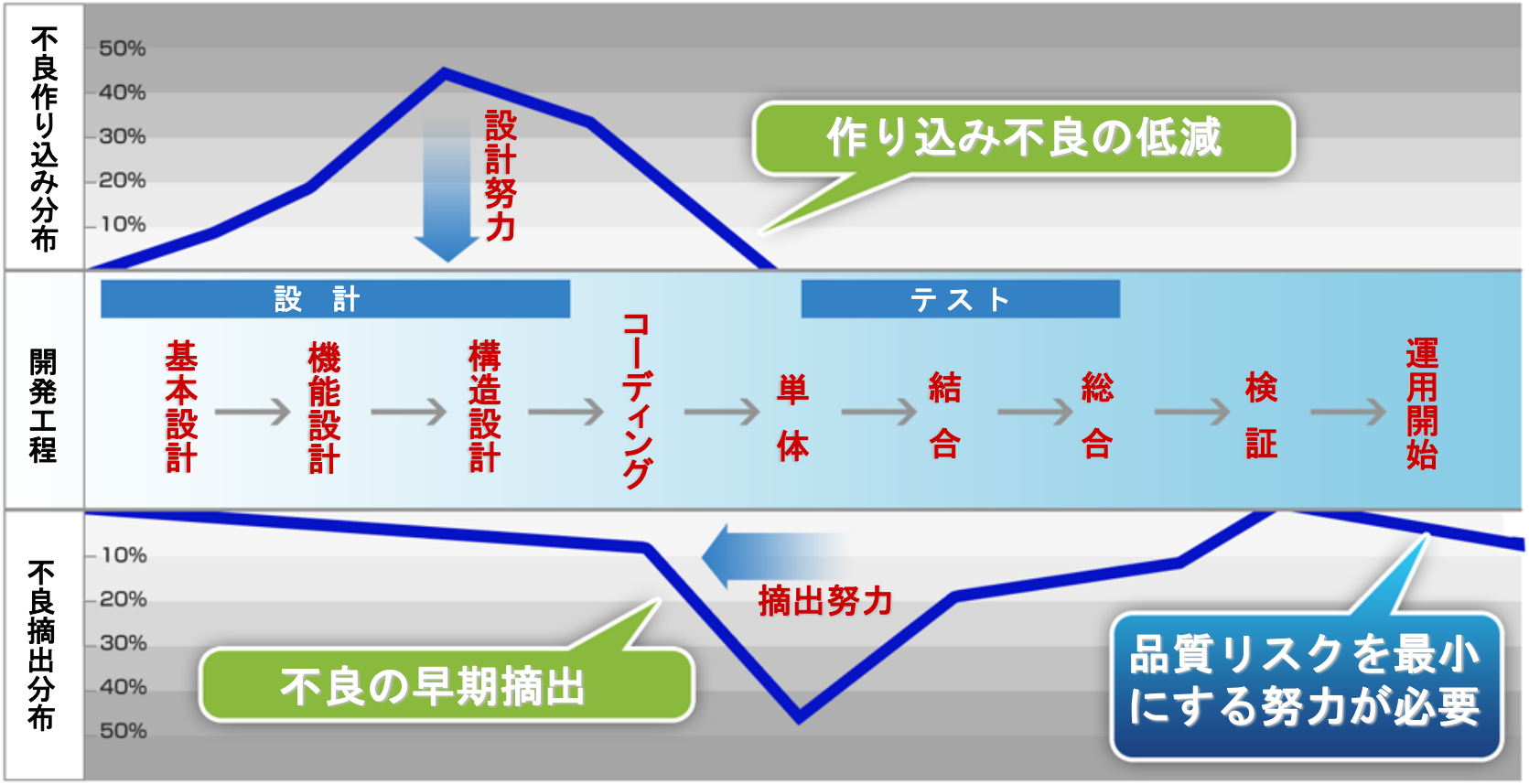
計測できないものは制御できない→トムデマルコ

【補足】V字モデル



V字モデル

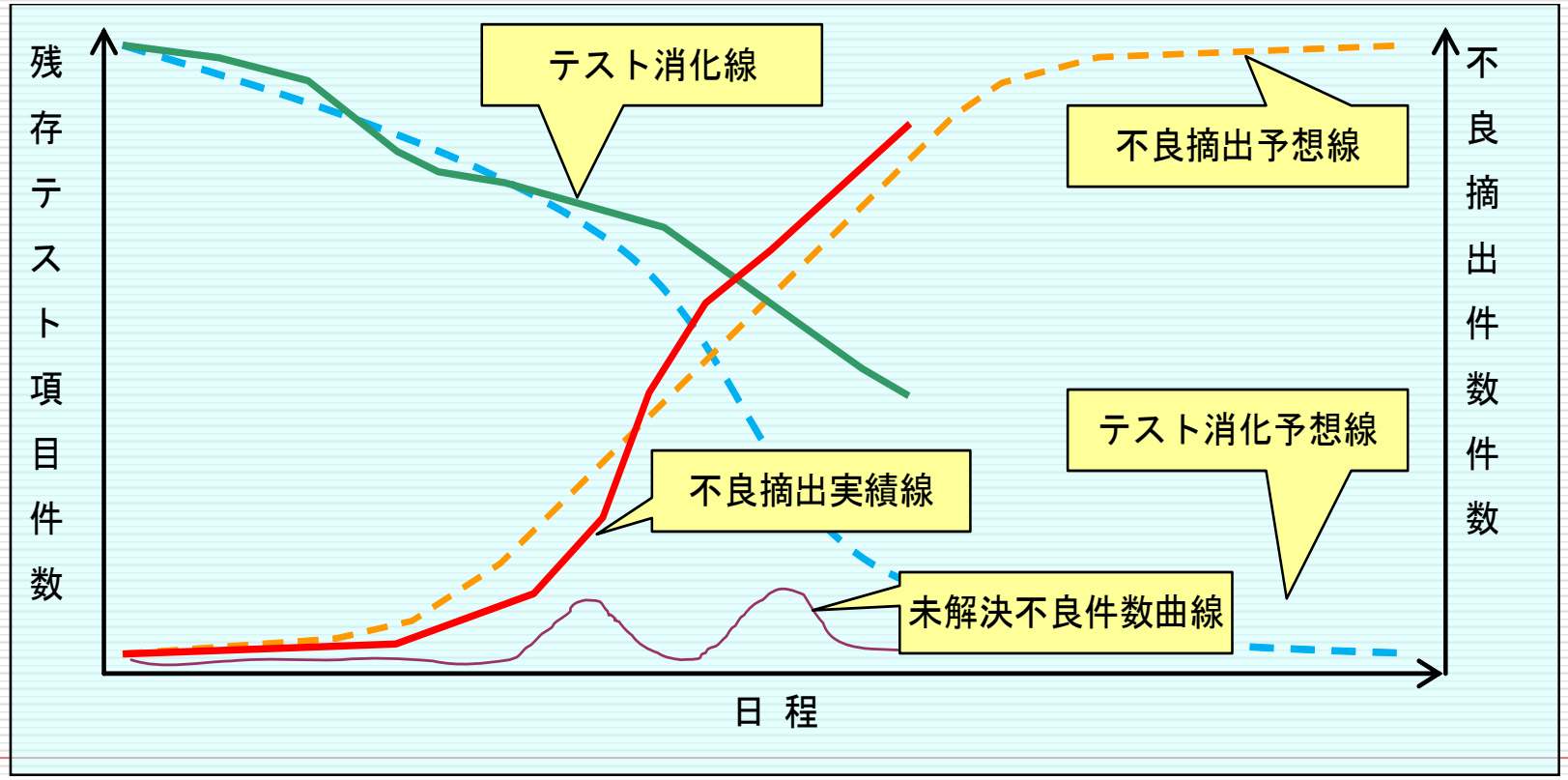
（1） 不良の作込みと摘出の関係



不良分布

（2） 品質管理図

テスト管理の代表的手法として品質管理図がある。
テストの進捗に併せ、テスト項目消化の予実績，障害発生・対策件数の推移をグラフ化し、傾向把握と状況管理を行う。



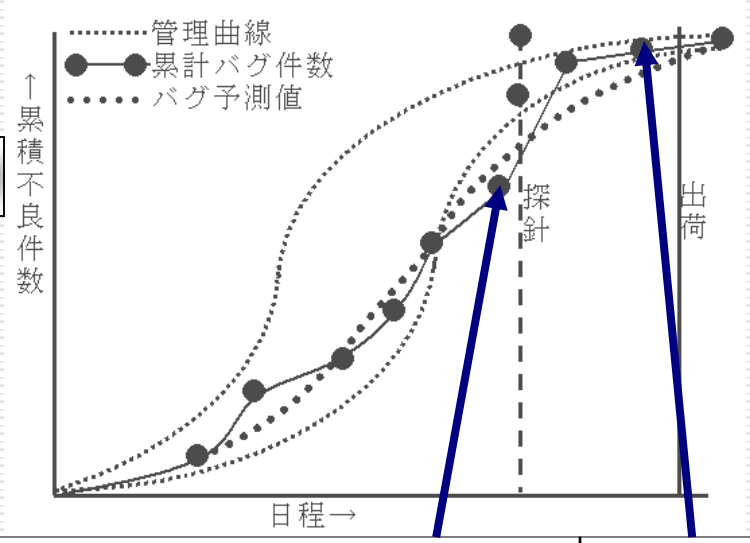
品質管理図

（3） 品質目標値管理の概念図

1. 品質目標設定

2. 管理曲線の設定

3. 実績の監視



6. 稼動管理

5. 品質改善

4. 目標値管理

異常値管理	収束状況管理
実績が管理曲線から外れた場合／原因究明と対策	収束率(*)による品質評価、管理

5. 品質予測と評価　信頼性成長曲線に基づくバグ件数の予測 ／探針（QP）

※収束率＝実績バグ件数／予測バグ件数

（４）探針（QP:Quality Probe）

- 探針は、検査項目をサンプリングし、その結果から全体の母不良率を求め残バグ件数を推定する方法である。

① 探針項目として、検査項目数（A）の10%～20%をサンプリングする。

② 探針項目から出た不良に対して不良率を次の式で求める。

標本不良率：$\rho' = r/n$ （n：探針項目数　r：不良件数　）

③ 二項確率紙または次の近似式により母不良率の信頼限界の上限値と下限値を求める。

※　危険率：αは通常5%としている。

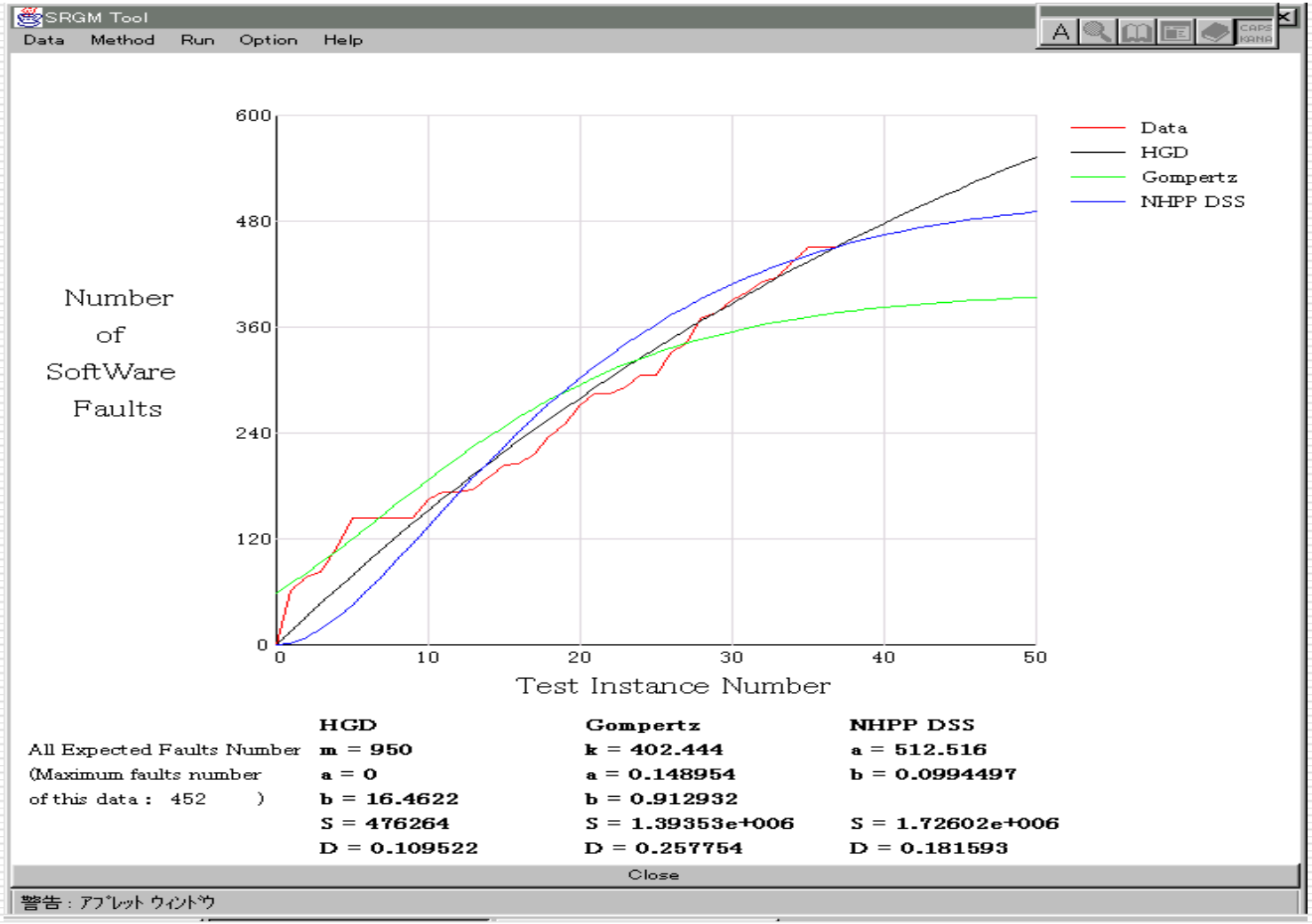
$u(\alpha)=1.9$

上限値：$P_U = \rho' + u(\alpha)\sqrt{\rho'(1-\rho')/n}$

下限値：$P_L = \rho' - u(\alpha)\sqrt{\rho'(1-\rho')/n}$

④ 残バグ数＝ $(P_L \times A) \sim (P_U \times A)$

（5） 信頼性成長モデル（具体例）



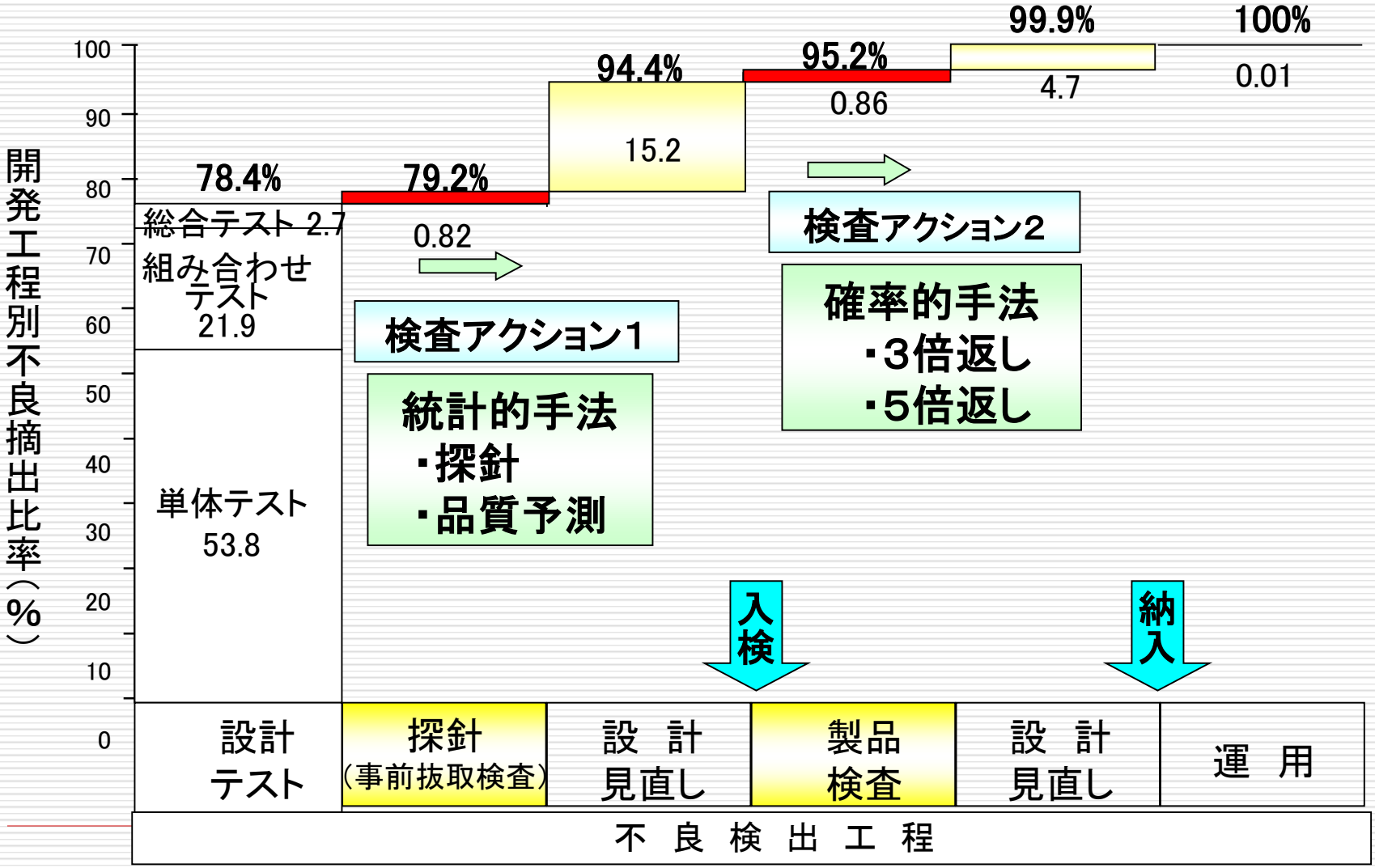
【補足】目標値設定（テスト工程）の具体例

ﾊﾞｸﾞ摘出／テスト項目設定目標値の例（KLOCベース）→80年代

単　体　テ　ス　ト	結　合　テ　ス　ト	総　合　テ　ス　ト
1.PCL件数： 100件／kloc 2.PCL質的内容 正常　：70%以下 異常　：10%以上 限界/境界：15%以上 ｲﾝﾀﾌｪｰｽ　：5%以上 3.バグ摘出件数： 8～10件/ kloc	1.CCL件数： 20～25件/ kloc 2.バグ摘出件数： 1～2件/ kloc	1.SCL件数： 5～10件/ kloc 2. バグ摘出件数： 0.5件/ kloc

＊ＰＣＬ：単体テスト項目、ＣＣＬ：結合テスト項目、 ＳＣＬ：総合テスト項目

【補足】開発工程と不良摘出



工程別不良摘出比率

3.3 1980年代品質保証活動③

＊＊モニタリング中心、当該プロジェクトへの
フィードバックが中心＊＊

＊＊プロセスへのフィードバックの概念は希薄　＊＊

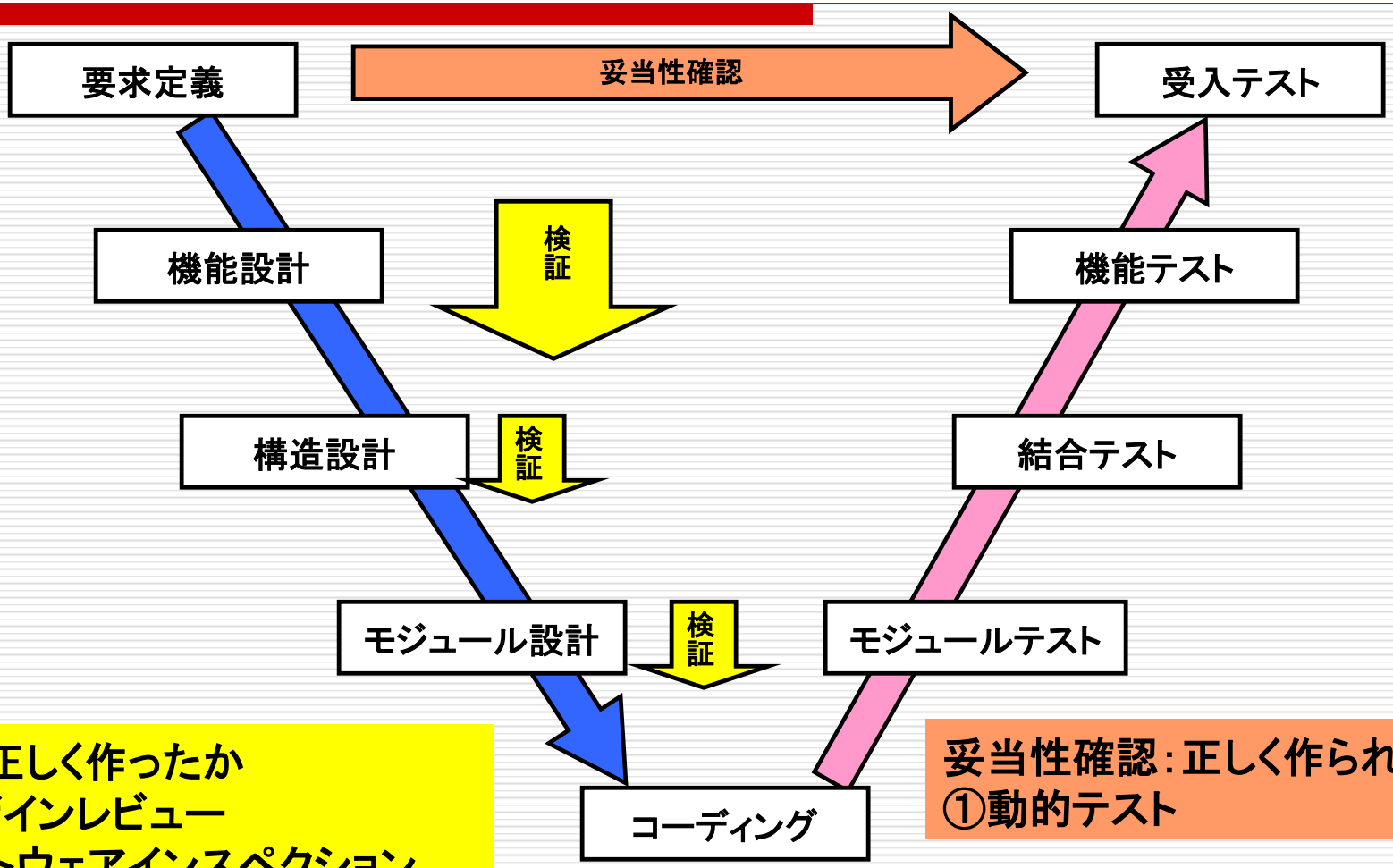
＊＊品質管理の隆盛期、現在の手法はこの時代に
確立された　＊＊

＊＊これが現在に継承され、進化しているか（？）＊＊

3.4 1980年代品質保証活動④

- ソフトウェア開発管理の体系構築
  - 70年代の経験を開発標準として定義
  - 要求定義から総合テストまでの一通りのプロセスの定義
    - 局面に対応したプロセスの詳細定義（開発部門が主体）
    - 品質改善プロセスの採用
  - V&Vに基づく品質確認プロセス採用
    - 各種レビューの適用、定着

【補足】V＆V（検証と妥当性の確認）



検証：正しく作ったか
①デザインレビュー
②ソフトウェアインスペクション

妥当性確認：正しく作られたか
①動的テスト

V＆V　（verification＆validation）

KYASUDA Copyright (c) 2005 TIU

（1）　デザインレビュー

- 公式レビュー
  - ソフトウェア開発工程の各マイルストーンで行う公式のレビュー。開発の達成度を審査し、次工程に進むことができるか判定する
  - ポイント
    - 管理面、技術面、プロジェクトの進捗などを審査し、合否を出す
    - PJの利害関係者（管理者、品質保証担当、PM,SE,etc.）参加
- 色々なレビュー

・ウォークスルー（ティーム内レビュー）

・ピアレビュー（ティームメンバー間レビュー）

・コードレビュー

・その他もろもろ

3.5 1980年代品質保証活動⑤

- 開発支援環境の整備
  - 支援部門（QA、PM，生産技術）の強化
    - 開発部門の自由度、自主性のはく奪の始まり？
  - CASEツールの開発、試行、適用、発展
    →開発環境の重要性を認識

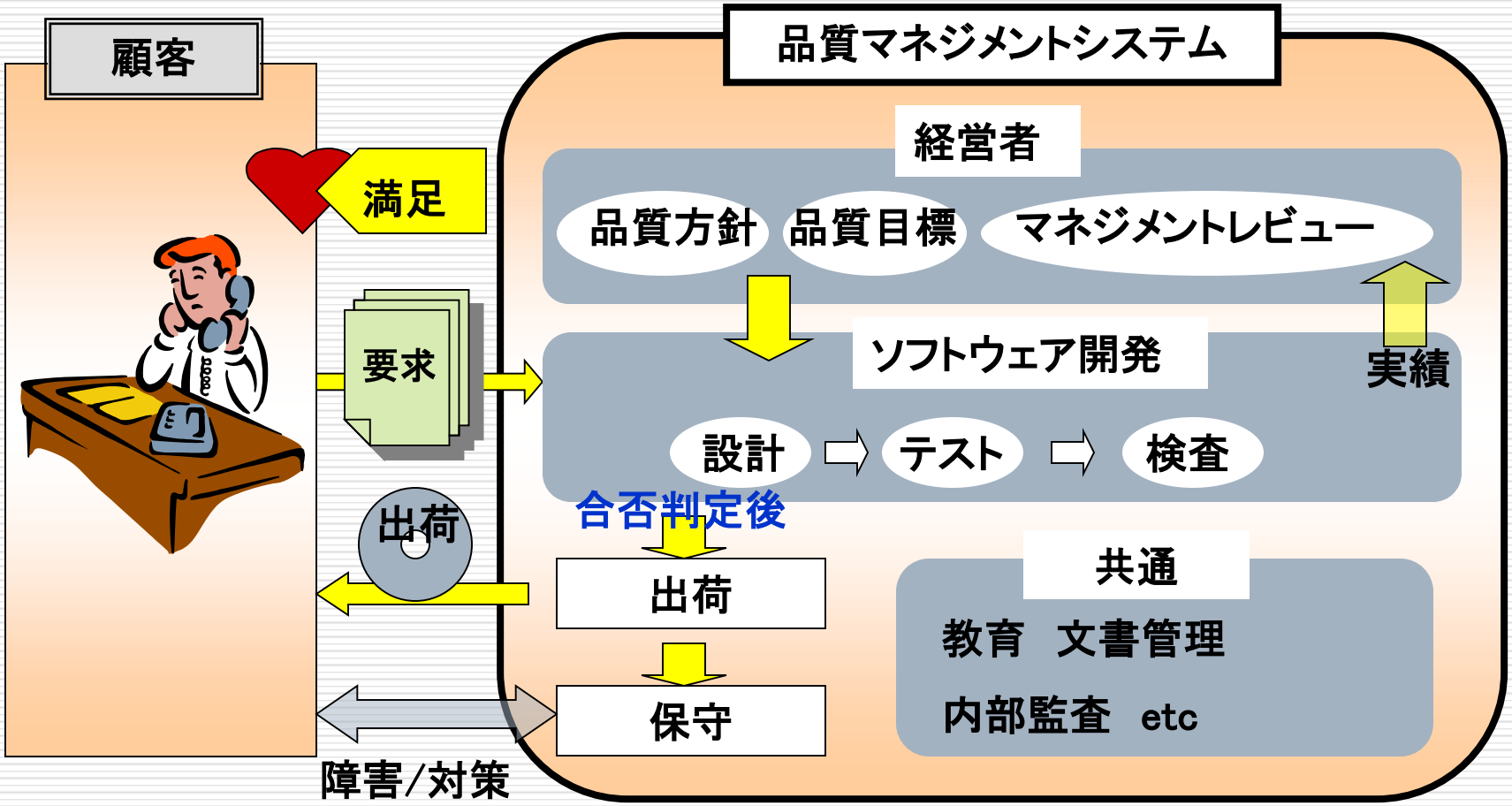
4.1 1990年代(前半)品質保証活動①

- 背景:品質管理の閉塞感とISO9000の台頭
  - 良いプロセスが良いプロダクトを生み出す → SPI の時代の開幕
  - プロダクト重視からプロセス重視へ

- ISO9000に基づく品質システム(客観的プロセス)の構築
  - 現場がプロセスの真意を理解できず形式的な取組み
    →品質システムの形骸化
  - 認証が目的化
  - プロセス定義、改善が開発部門から支援部門の手に
  - プロセス監査の導入→現場監視が目的、実施タイミング遅れにより
    形骸化

＊＊＊プロセスは専門家が作るとの悪しき習慣化＊＊＊

【補足】ISO9000の概要

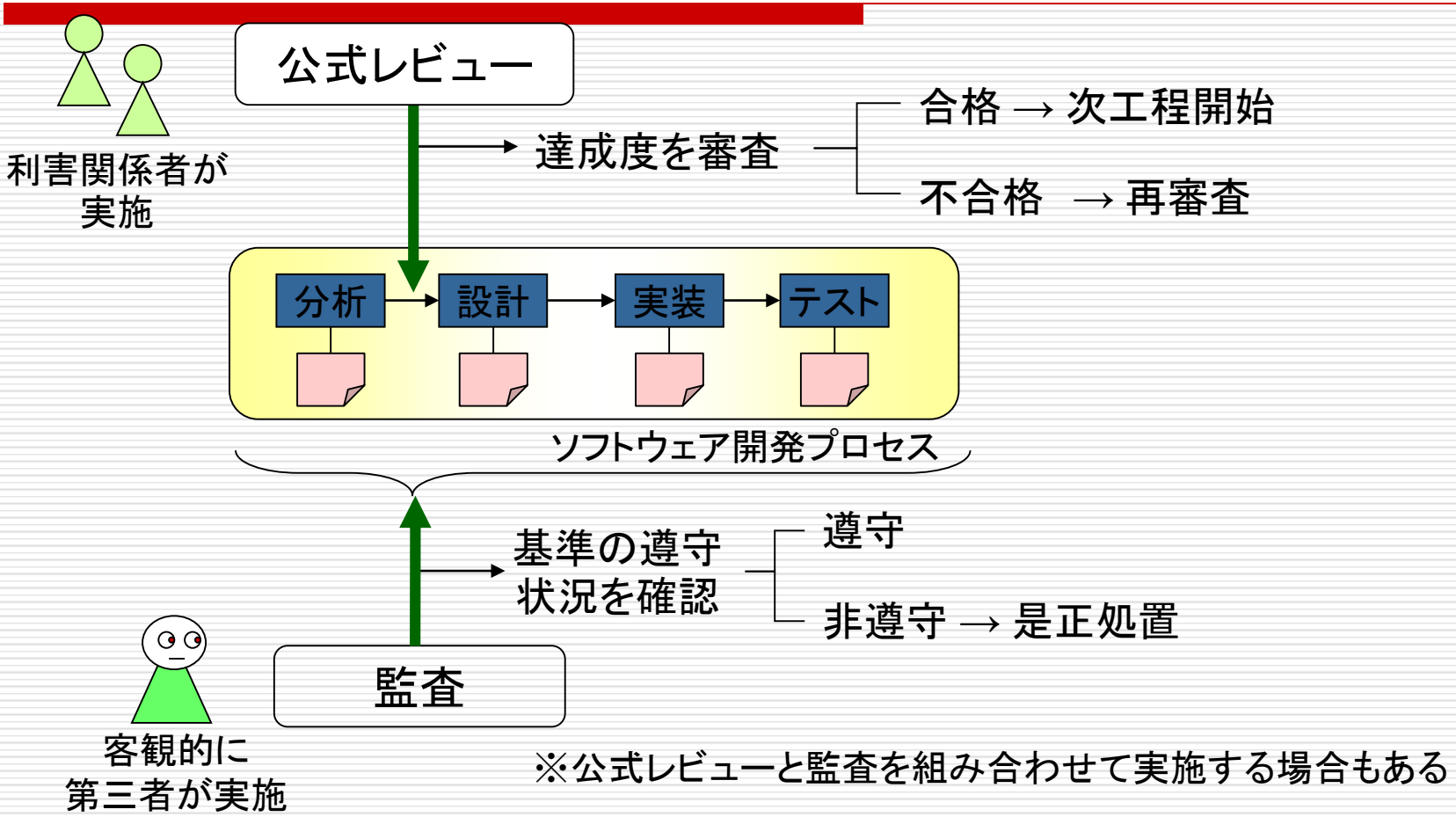


(1)監査とは（1/2）

- 「監査基準が満たされている程度を判定するために、監査証拠を収集し、それを客観的に評価するための体系的で、独立し、文書化されたプロセス（ISO9000:2000）」

- ポイント
  - ・基準（プロセス、ルール、標準 etc.）の遵守状況を評価する
  - ・客観性をもって実施
  - ・適切なタイミングで実施
  - ・フィードバック（プロジェクト、標準プロセス）

(1) 監査とは(2/2)



※公式レビューと監査を組み合わせて実施する場合もある

- 品質作り込みのプロセスが機能していることの確認が重要

スラド提供：東芝　小笠原秀人氏

4.2 1990年代（前半）品質保証活動②

- SW－CMM発表
  - ➡ ISO9000に比べプラクテスが明確、枠組みがスマート
  - ➡ 適用にあたり、専門知識が必要

- **プロセスの自由度が開発部門から更に剥脱？**
  - **改善モデル（例：IDEAL）が提示されたが普及せず**
  - **プロセス　VS　プロダクトの不毛な対立が勃発**

＊＊アセスメントは成熟、改善のスキームが未成熟＊＊

＜＜SPIは‘80年代の品質管理技術の上に成り立つ＞＞

5.1 1990年代(中半～)品質保証活動①

□ 背景:オープン化時代の到来、ITバブルの崩壊

- 開発スピード重視→ビジネスチャンス重視
- 作るから使う（組合せ）へ、グッドイナッフ、
  短サイクル/短納期、ベンダーの多様化、
  製品のブラックボックス化

➡ 開発現場の混乱、開発部門の自信喪失

➡➡ そして日本的品質管理の崩壊

➡➡ 新たな管理技法の模索

5.2 1990年代(中半～)品質保証活動②

- プロジェクトマネーメン(PM)の採用
  - モダンプロジェクトマネージメントの導入（例：PMBOK）
    - ➡ KKDからの脱却
      - ➡ 手法未成熟のため表面的、形式的適用で成果が上がらず
        - ➡ 現在この状況は脱却出来ているか？
  - 当時の誤解

    ①技術的問題をマネージメンとで解決しようとした。
    ②PMP取得が全てを解決する
    ③PMrに全てを押し付け、PMrの孤立化

- プロジェクトマネジメントシステム(PMS)の構築と試行
  - 現場をモニタリング、制御する仕掛けのため成果出ず
  - 現在のPMSの基礎の確立

【補足】オープン化に伴う品質管理の課題

＊課題解決に直接手を出せない
　⇒解決に指向錯誤を繰り返す

＊スコープが不明確（エンドユーザ指向）
　⇒不採算プロジェクトが多発する

＊システムは一発勝負、短納期である
　⇒MFのように次プロジェクトで挽回は不可

プロジェクトマネジメント的な対応、手法が必要
⇒リスクマネジメント、スコープマネジメント、組織マネジメント

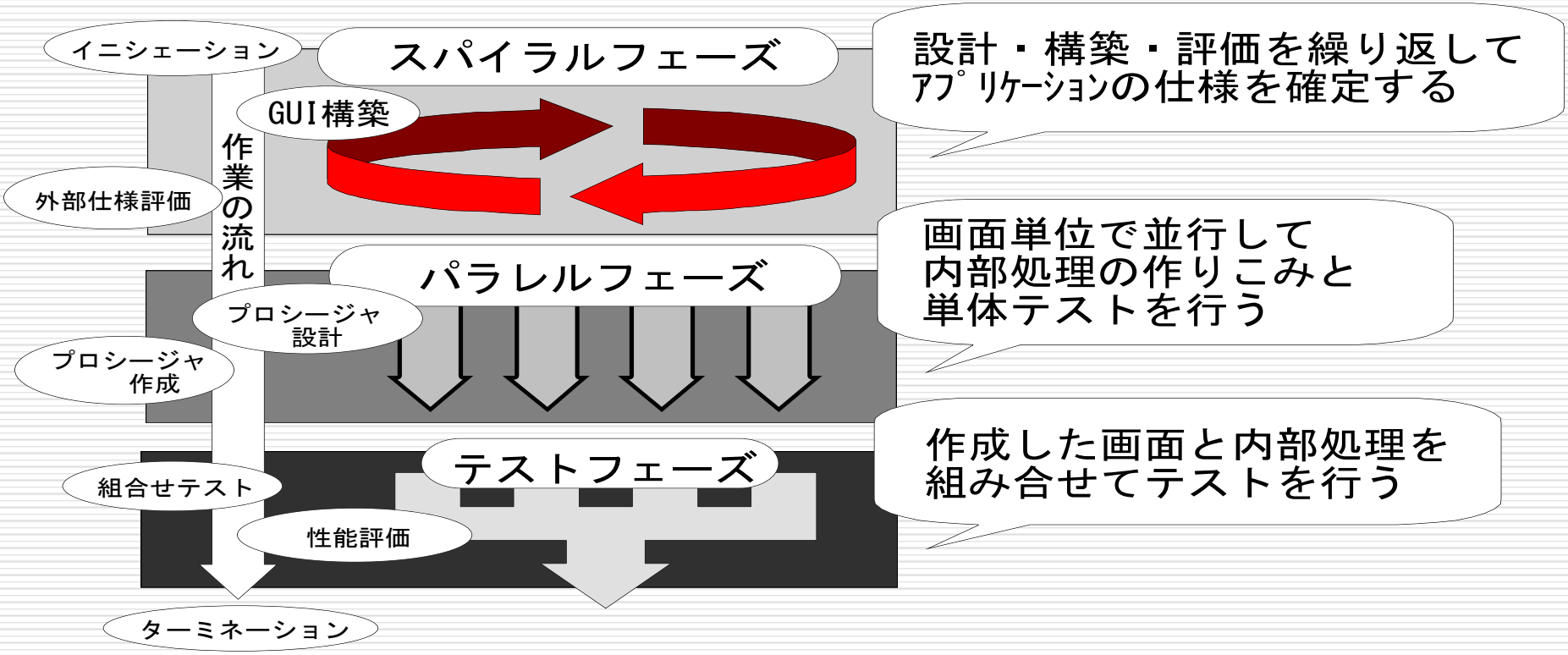
5.3 1990年代(中半～)品質保証活動③

- パッケージソフトの最適組み合わせノウハウの蓄積、活用
  - プロトタイプ的な思考、錯誤
  - PKG間の相性、親和性（非技術的（？）表現が）　課題に

- 多様な開発環境の出現(提案)→簡単、短時間がキーワード
  - 銀の弾丸と錯覚させられた？
  - 一過性のものが多数出現

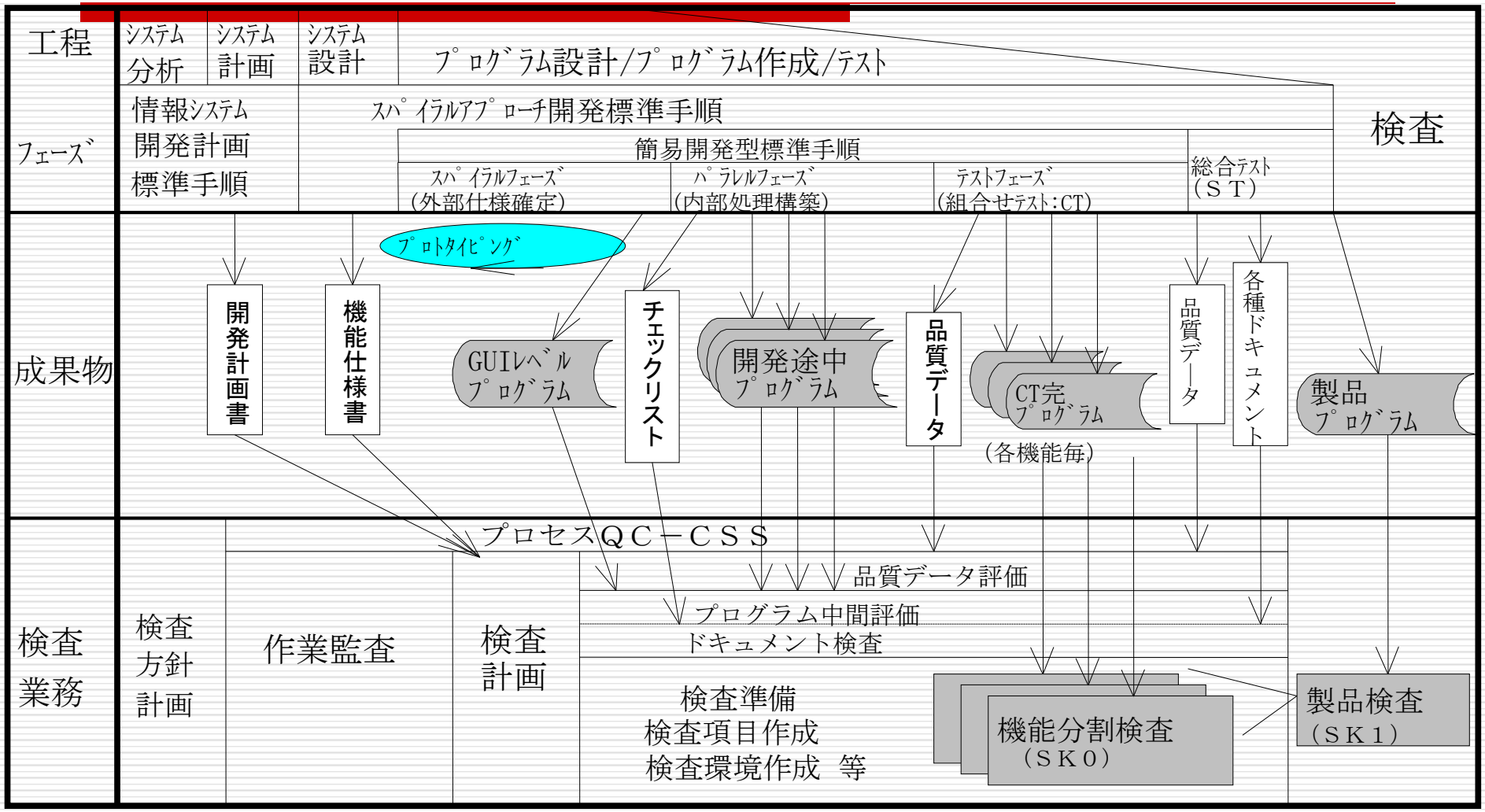
    →現在も生き残っているのは？？？

【補足】オープンシステムの開発モデル



オープンシステムの開発モデル

【補足】オープンシステムの標準検査プロセス



オープンシステム対応標準検査作業

【補足】目標値設定（テスト工程）の具体例

ﾊﾞｸﾞ摘出／テスト項目設定目標値の例（FPベース）→90年代

単　体　テ　ス　ト	結　合　テ　ス　ト	総　合　テ　ス　ト
1.PCL件数： 4.1件／FP 2.バグ摘出件数： 0.34件/FP	1.CCL件数： 1.0件／FP 2.バグ摘出件数： 0.04件/FP	1.SCL件数： 0.4件／FP 2. バグ摘出件数： 0.02件/FP

※この目標値は統計的手法により求めた値であり、各プロジェクトにおいてはその特性を考慮しプロジェクト毎に設定する必要がある。

＊ＰＣＬ：単体テスト項目、ＣＣＬ：結合テスト項目、 ＳＣＬ：総合テスト項目

6.1 2000年代　品質保証活動①

- 背景：日本経済の回復、日本ソフトウェアの危機感の喚起
  →ＩＴ構造改革の機運の芽生えと取組み
  → 品質保証への多様な取り組み機運の醸成

- モダンＰＭへの本格的な適用
  - 本格的ＰＭＳの開発と適用、ノウハウの蓄積と活用

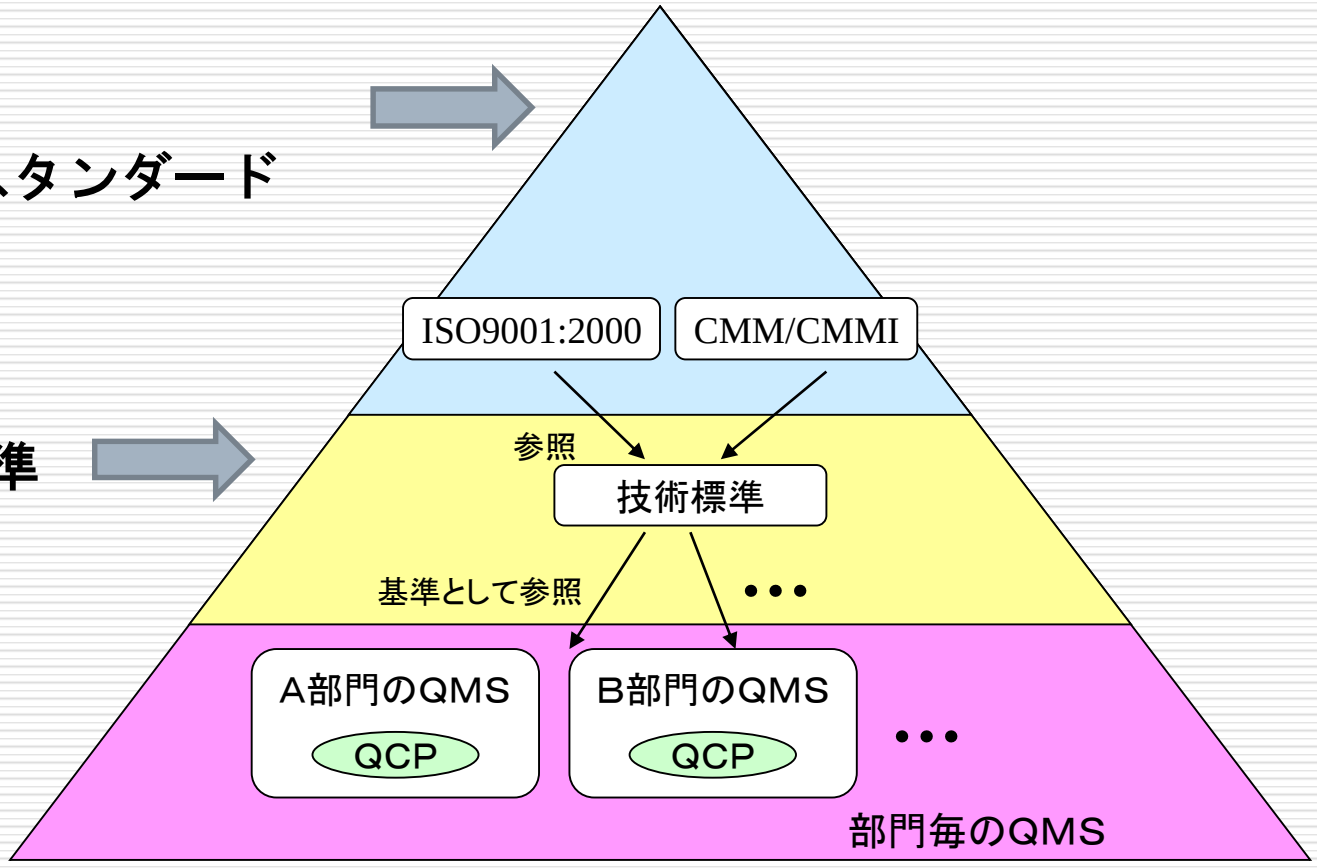
- SPIへの本格的な取り組み　→　認証取得からの脱却、PMSの再構築
  - ISO 9001-2000（品質マネージメントシステムへ）
  - ＣＭＭＩの発行を契機に、真に効果的適用方法、改善手法の模索
  - ＰＰＱＡへの本格的な取り組み
    →正しく作ったか、正しく作られたか、の両立性
    →プロセス　VS　プロダクトからの脱却

【補足】品質保証のフレームワーク

国際的な
デファクトスタンダード

社内共通の技術標準

各部門の
品質マネジメント
システム



プロセスの構造

6.2 2000年代　品質保証活動②

- 日本的品質管理の再評価とリニューアル適用
  - 社会的トラブル多発原因の評価結果が誘引
- ソフトウェアテストの重要性の再認識
  - 科学的、工学的な取組み強化
  - テストプロセス確立、普及（特にテスト設計の重視）
- Vモデル拡張版の提案と試行
  - ダブルＶ字モデル、Ｗ－モデル
- Ｖ＆Ｖ(検証と妥当性確認)の本格的適用
  - 検証の有効性の認知、ＳＷインスペクション、公式レビュー強化

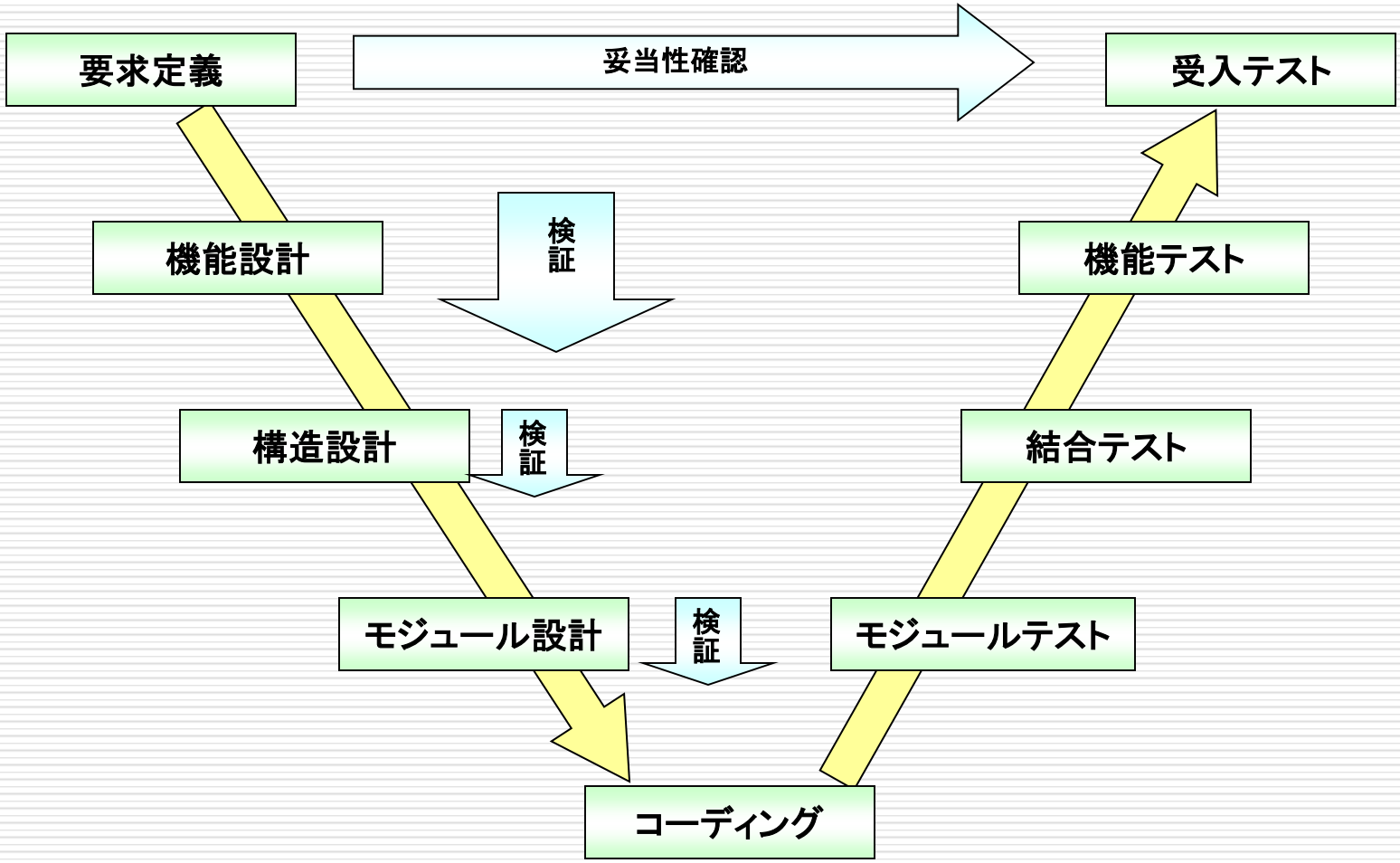
(1)日本的品質管理のリニューアル

1. 日本的品質管理の再評価とリニューアル適用

- ＴＱＣ的ＳＰＩ，品質改善の復活
  ⇒現場に根ざした改善活動の再構築
  ⇒改善に直結するメトリックスの活用

- 各種デザインレビューの適用
- 基幹3大メトリックス（バグ、規模、工数）の採取と活用
- V&Vモデルに基づく検証の強化
- テストにおける品質改善プロセスの採用

【補足】V＆Vに基づく検証と妥当性確認



検証の充実

【補足】検証の充実　～ソフトウェアインスペクション～

（1）ソフトウェアインスペクションとは

⇒対象の合否判定とプロセス改善

□　レビュー対象仕様書（成果物）と当該仕様書作成作業の入力資料（ソース文書）を比較しながらレビューを行う

□　ソフトウェア要素が、その仕様を満足していること、適用標準に従っていることの確認、標準および仕様からの逸脱が無いこと

□　ソフトウェア工学データの収集（例；工数、欠陥のデータ）

（２）実施方法

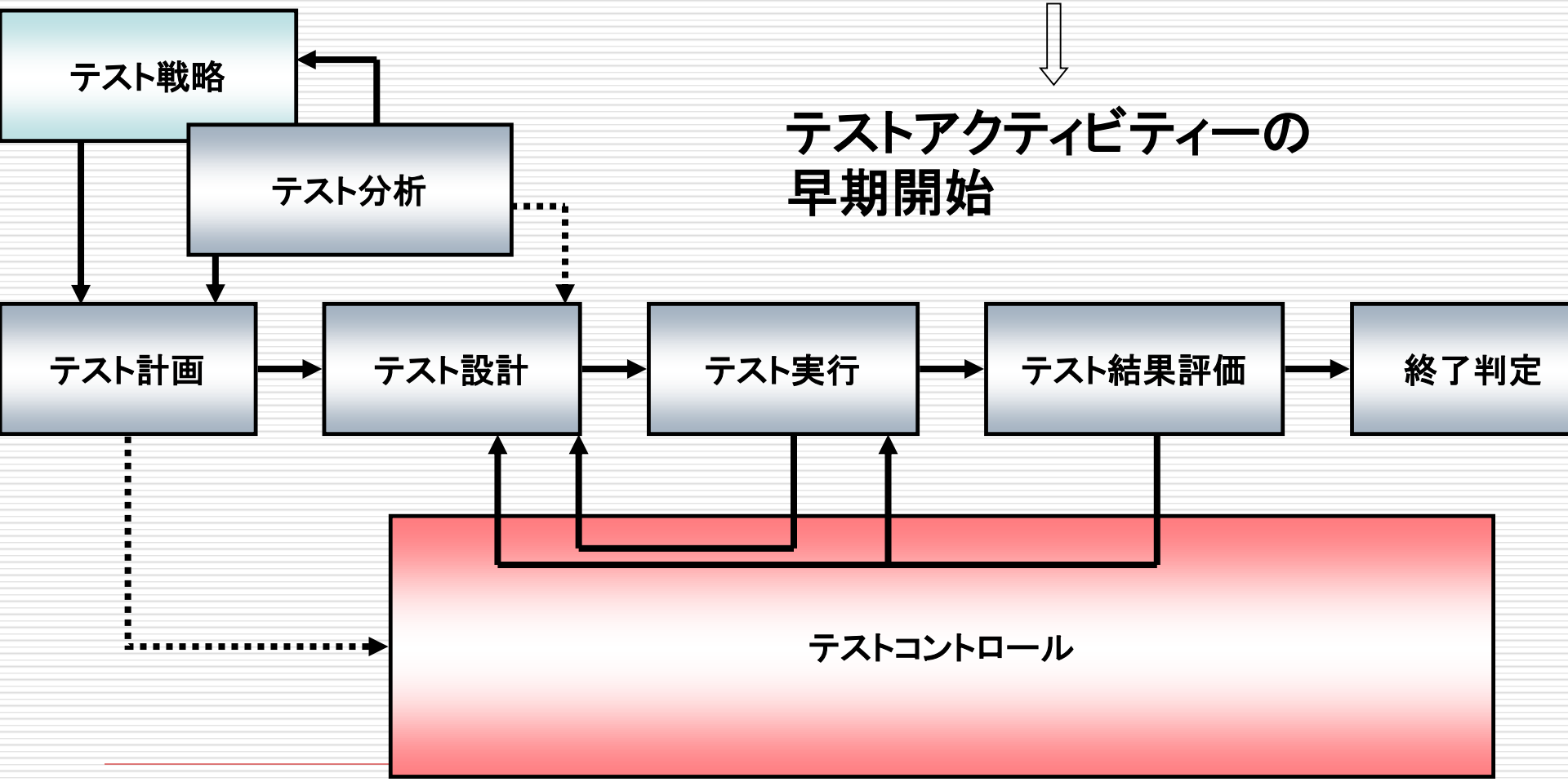
□　基本的な方法はウォークスルーと類似、但しルールは厳密

□　モデレータと呼ばれる実施責任が、会議を計画／実施、レビューアの選定、結果報告書、指摘事項の修正確認まで全てを取り仕切る

□　結果の収集分析と有効活用（エラーの分類、タイプの分類、エラーの記録、をして開発者にフィードバック、類似エラーの摘出を支援）

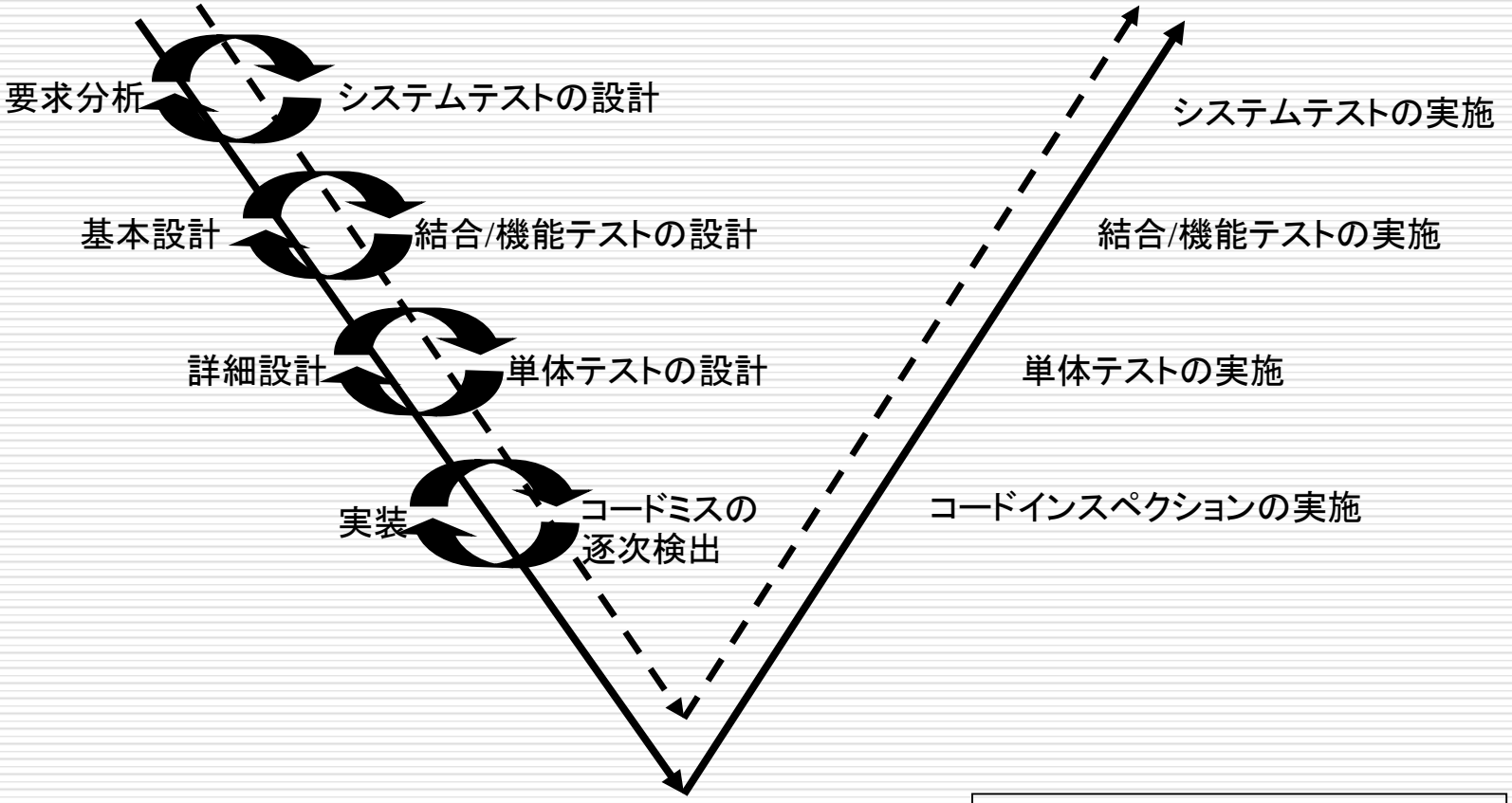
6.3 ソフトウェアテスト重要性の再認識

テストプロセスの定義



テストプロセス

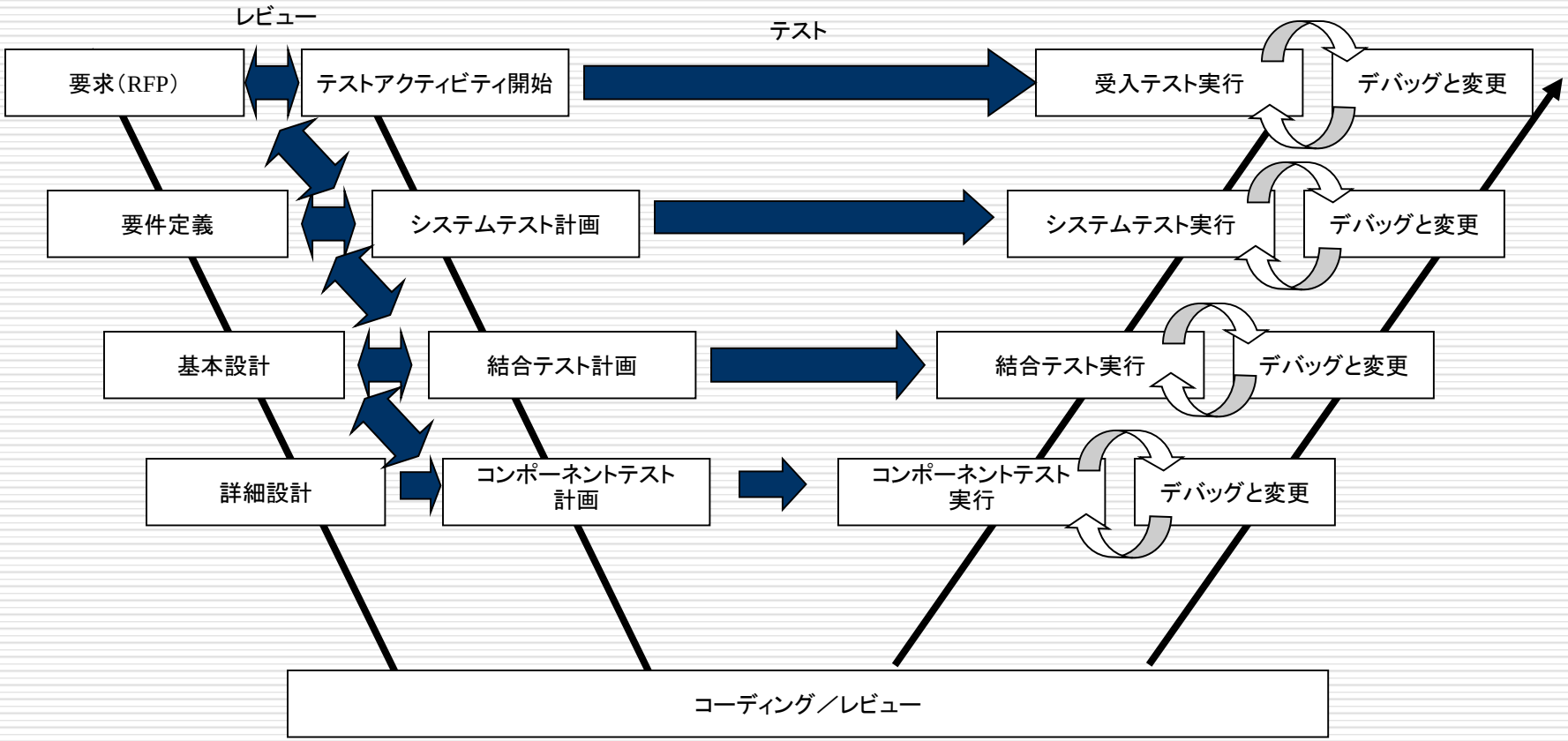
【補足】ダブルV字モデル



電通大　西先生　講演資料より

ダブルV字モデル

【補足】開発のプロセス：W-model



（抜粋）出典：JaSST’08TokyoチュートリアルTsuyoshiYUMOTO

6.4 日本的管理技術の伝承（積木崩しからの脱脚）

ソフトウェア品質の測定とメトリックス

（１）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**必要不可欠。**

⇒結果の活用＝プロセスへの組込み（フィードバック）

（２）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継らない。

⇩

現在も起きている最大の課題では！！

(1) ソフトウェア品質の測定とメトリックス

- **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と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？

(2)ソフトウェア測定の考え方とメトリックス

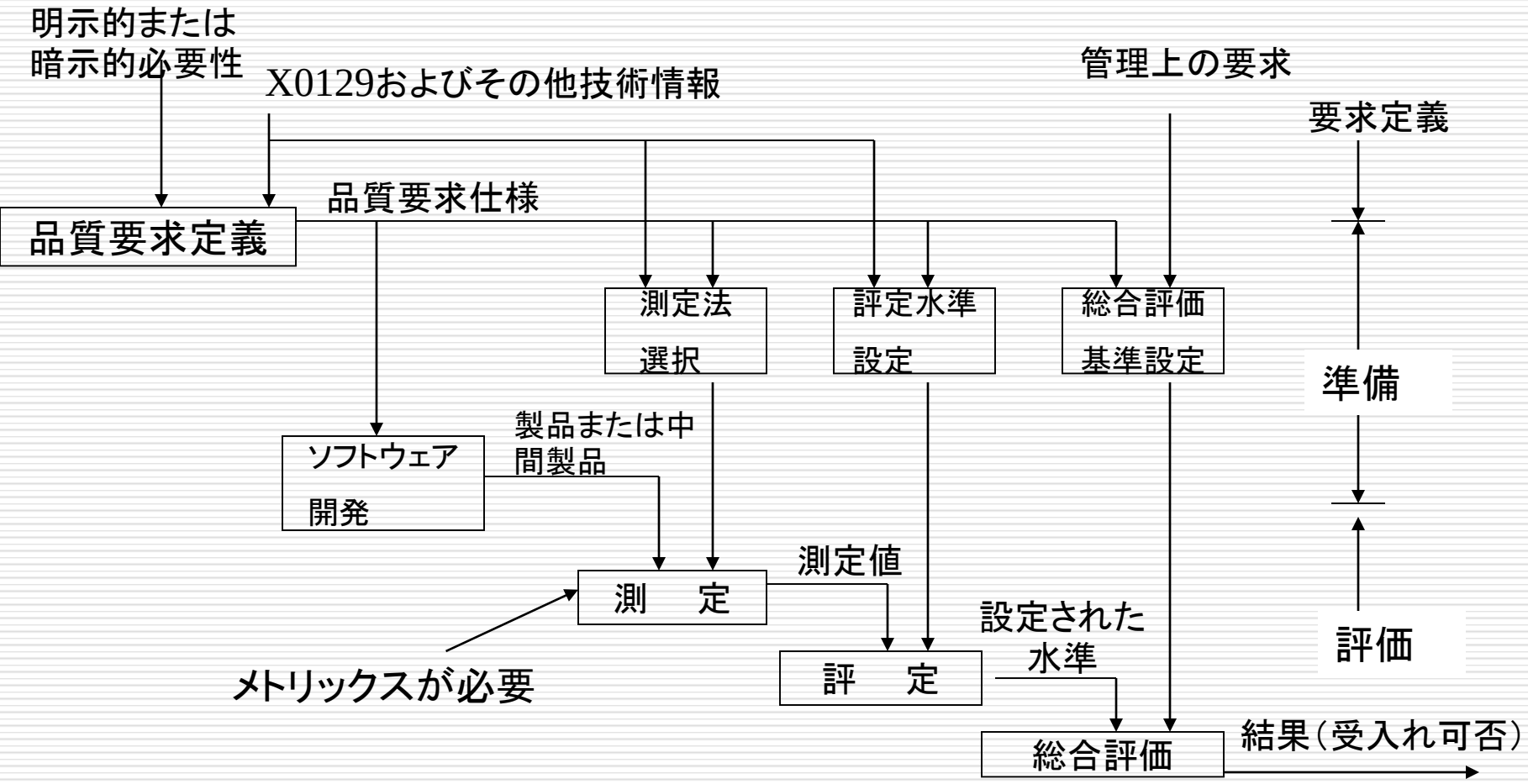
- 計測は品質管理、プロジェクト管理のみならず、あらゆる管理(制御)の基本である
  - 「測れないものは制御できない（トム．デマルコ）」
- 計測を確実にするために
  - 計測が容易であること
    - 計測の容易（可能）な尺度を選択
    - 計測を容易にする仕掛けとツールの導入
  - 計測実績データの蓄積．評価
    - 計測条件（コンテキスト）の明確な実績データの収集．蓄積
      - 計測条件が異なるデータ（絶対値）の比較は意味が無い
    - 実績データの分析．評価
    - 評価基準の設定

【補足】メトリックスの定義

- “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
- 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て、改善することが重要
  - どの様なメトリックスを用いるかは、何を目的にどの様な目標を設定するかで決定

（　出展：ＳＱｕＢＯＫガイド）

【参考】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。

ソフトウェア測定（まとめ）

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。

④ 測定のためには、最低限のソフトウェア工学の知識が必要である。

計測事象＝因果関係＋バイアス＋偶発性

（出展；長崎大中村先生ーJASPIC講演より）

【蛇足】古典的な３大メトリックス

- 開発規模
- 開発工数
- バグ数

ー　未だに取り続けているが有効に活用されているか疑問

ー　未だに取っていない組織がある、驚き！

ー　SEI(CMMIティーム）はコアメトリックスと呼んでいる

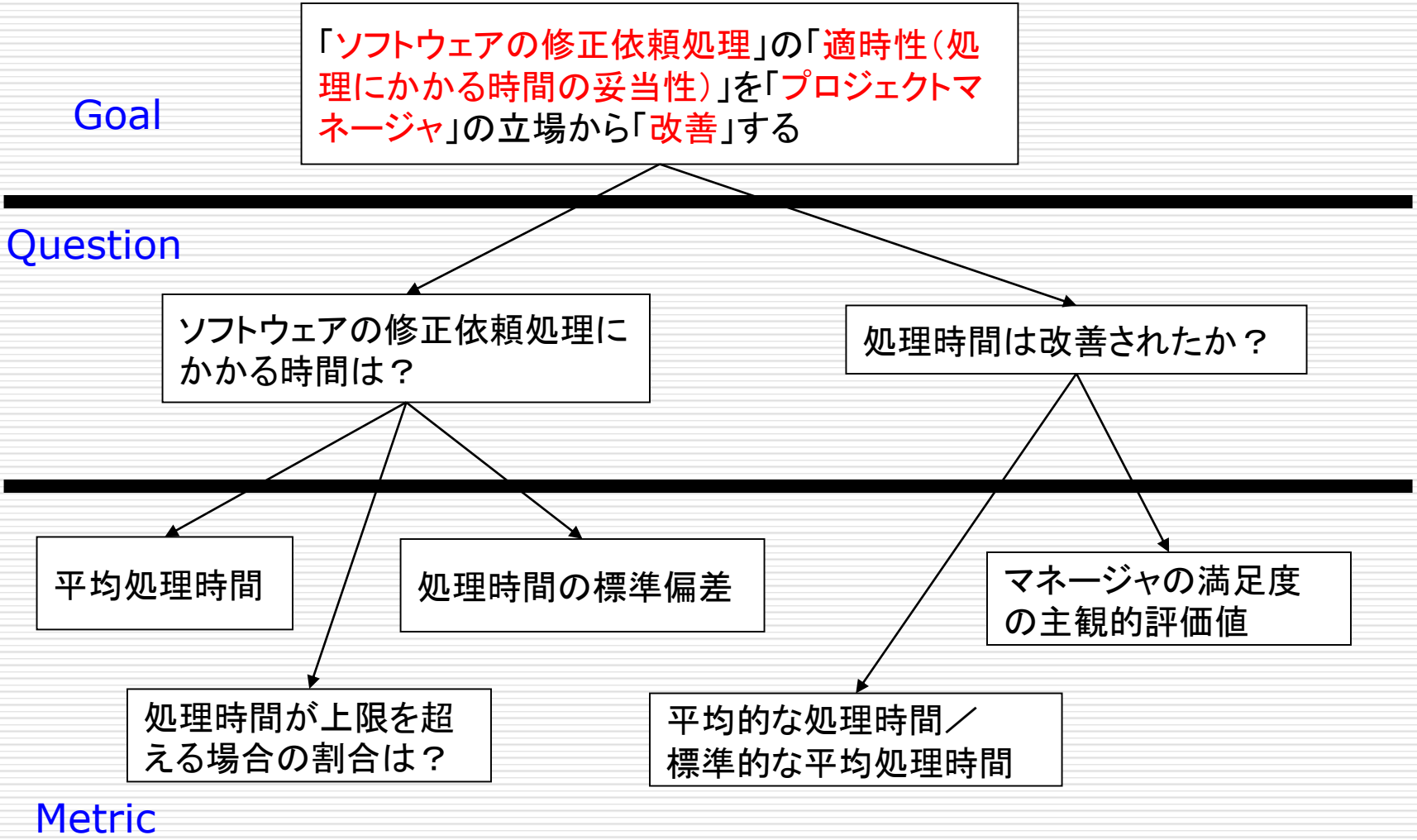
～　新しいことのように言っている、惑わされてはいけない　～

【参考】メトリッ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.

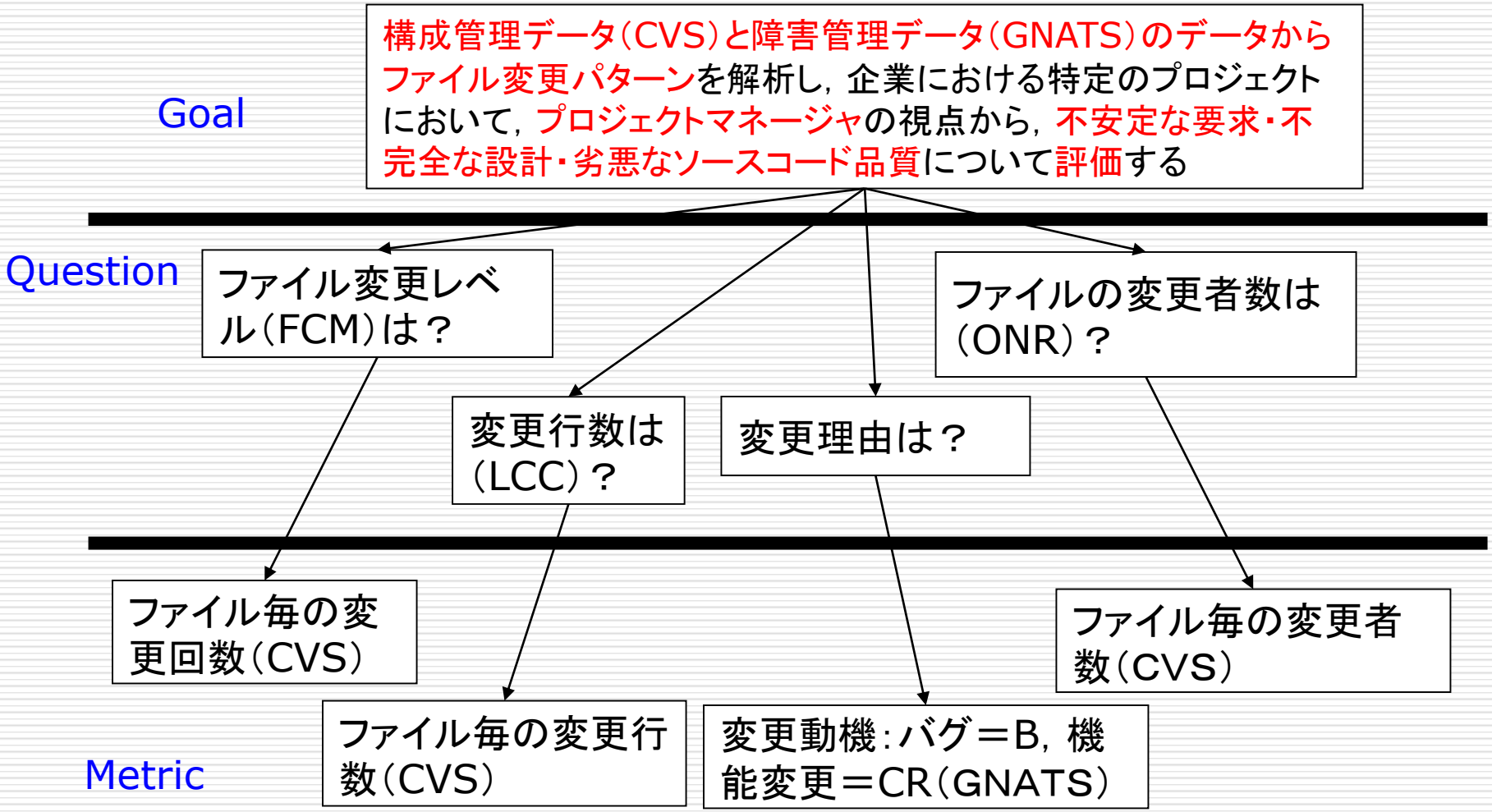
出展：EASEプロジェクト

GQMモデル例*



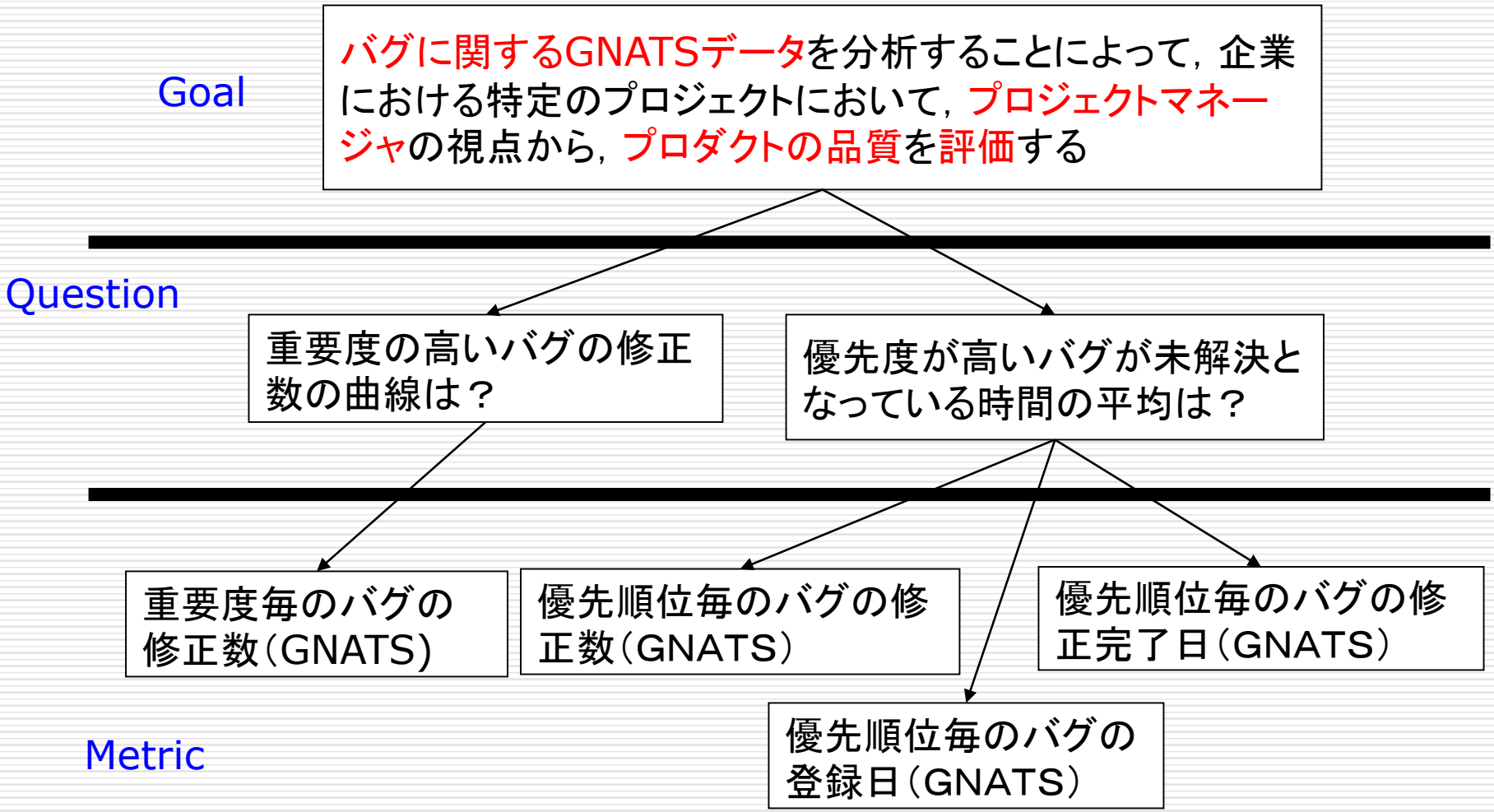
*井上克郎，松本健一，飯田元　著「ソフトウェアプロセス」共立出版，2000.

GQMモデル（1）



出展：EASEプロジェクト

GQMモデル(2)



出展：EASEプロジェクト

【参考】 ソフトウェア工学の基礎知識の習得

（1）ソフトウェア工学の基礎理論・知識

開発エンジニアは、本来「何をしなければ」成らないのか、「何が出来なければ」ならないのかという、規範を再確認し、その学習の基本となる「ディシプリン」を何に求めるかも再認識する必要が有る。

これの一つの例としてパーソナルソフトウェアプロセス（PSP)がある。

＊＊＊＊＊ PSPが定義している学習事項 ＊＊＊＊＊＊＊＊

① 時間管理 ②時間の追跡 ③期間計画と製品計画 ④製品計画
⑤製品規模 ⑥時間の自己管理 ⑦コミットメントの管理
⑧スケジュール管理 ⑨プロジェクト計画 ⑩ソフトウェア開発プロセス
⑪欠陥 ⑫欠陥の発見 ⑬コードレビューチェックリスト
⑭欠陥の予測 ⑮欠陥除去の経済学 ⑯設計欠陥 ⑰製品品質
⑱プロセス品質 ⑲品質に対する個人的コミットメント

まとめ①

- ✔ 品質保証とは、ユーザに「安心感」を持って頂くために開発者、提供組織が行うべき諸活動（アクティビティー）。アクティビティーは全プロセス、全組織、人、文化の多岐に渡る。
- ✔ 開発者の全ての仕事は品質保証に直結している。
- ✔ 品質保証とは「特別な作業」を実施することではない。基本（遣るべきこと）を守ることこそが最も重要である。
- ✔ 開発における「測定技術」はコアコンピタンスである。「測定」にはソフトウェア工学の知識が必要になる。
- ✔ 品質保証技術の改善、発展にはその変遷を知る事が重要。従来の技術を咀嚼しそれにアイデアを積重ねるのが真の技術発展になる。

まとめ②

＜＜実行委員の最終質問＞＞

Q　;講演者に取って「品質保証活動」って何ですか？

A　;自分の仕事および結果（成果物）に責任を持つこと。
そのため;
①自分の仕事の足跡を時々振返り、齟齬のないことを確認する。
②次のステップに踏み出す時に、自分が遣ろうとしていることを再確認する。
③時々自分の仕事の棚卸しを行い、プラクティスやプロセスを見直し、最適化を検討する。

温　故　知　新

そして

温　顧　創　新

ご清聴ありがとう御座いました