

JASPIC SPI Japan 2009

(2009-10-5 新潟朱鷺メッセ)

ソフトウェア品質保証の 方法論、技法、その変遷 ～先達の知恵に学ぶ～

2009年10月5日

NARAコンサルティング

奈良 隆正

今日お話すること

- ソフトウェア(SW)品質／品質保証の定義など
- 70年代のSW品質保証技術と活動
- 80年代のSW品質保証技術と活動
- 90年代前半のSW品質保証技術と活動
- 90年代後半のSW品質保証技術と活動
- 2000年代のSW品質保証技術と活動
- ソフトウェア品質保証の再生に向けて

品質保証の定義(現在)(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

*** 詳細は“SQuBOKガイド”参照 ***

ソフトウェア品質知識体系ガイドとは？

□ SQuBOK[®]ガイド

Guide to the **S**oftware **Q**uality **B**ody **o**f **K**nowledge



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)

品質保証の定義(過去)

- 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—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)

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)

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:

- 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。
- 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。
- 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>>

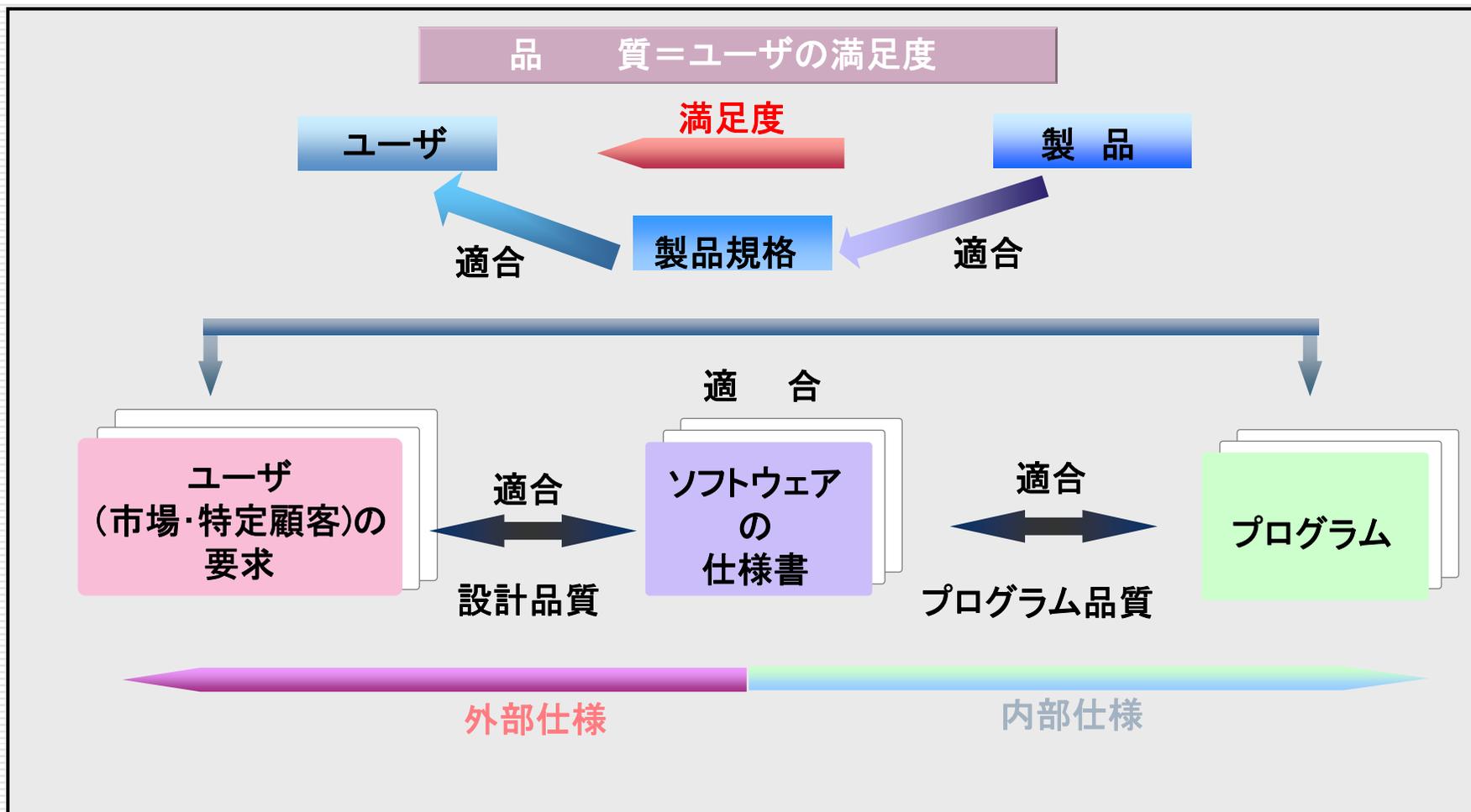
品質保証は品質保証部門の仕事であり、そこが頑張っ
てやるもの。

- 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」重要視して発展(契約社会という文化的な背景がある)

ソフトウェア品質の考え方

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。



<ソフトウェア品質 二つの条件>

- 1、設計仕様がユーザ要求に合致していること。
- 2、プログラムが設計仕様を実現化し、仕様通りに正しく動作すること。

設計品質は外部仕様の解釈により定まり、
プログラム品質は内部仕様の正確性によって定まる。

さらに最終成果物のプログラムが
顧客の要求を満たしていること。



品質の定義の変遷(1/3)

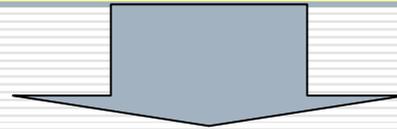
JIS(JIS)Z8101 品質管理用語)およびISO8402では、品質を以下のように定義している。

●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。」

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

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

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	<p>＜論理の集合であること＞</p> <ul style="list-style-type: none">・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	<p>(1)構造設計とそれに基づくレビュー (2)システムマチックなテスト</p>
2	<p>＜目に見えないこと＞</p> <ul style="list-style-type: none">・品質管理が困難・工程管理が困難	<p>(1)品質のビジュアル化 (2)開発工程のビジュアル化</p>
3	<p>＜個人への依存が高いこと＞</p> <ul style="list-style-type: none">・個人差が大きい・多人数の共同作業	<p>(1)開発手法の標準化と自動化 (2)再利用技術 (3)教育</p>
4	<p>＜ユーザニーズと直結していること＞</p> <ul style="list-style-type: none">・ユーザニ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)	<p>(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)</p>

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

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)

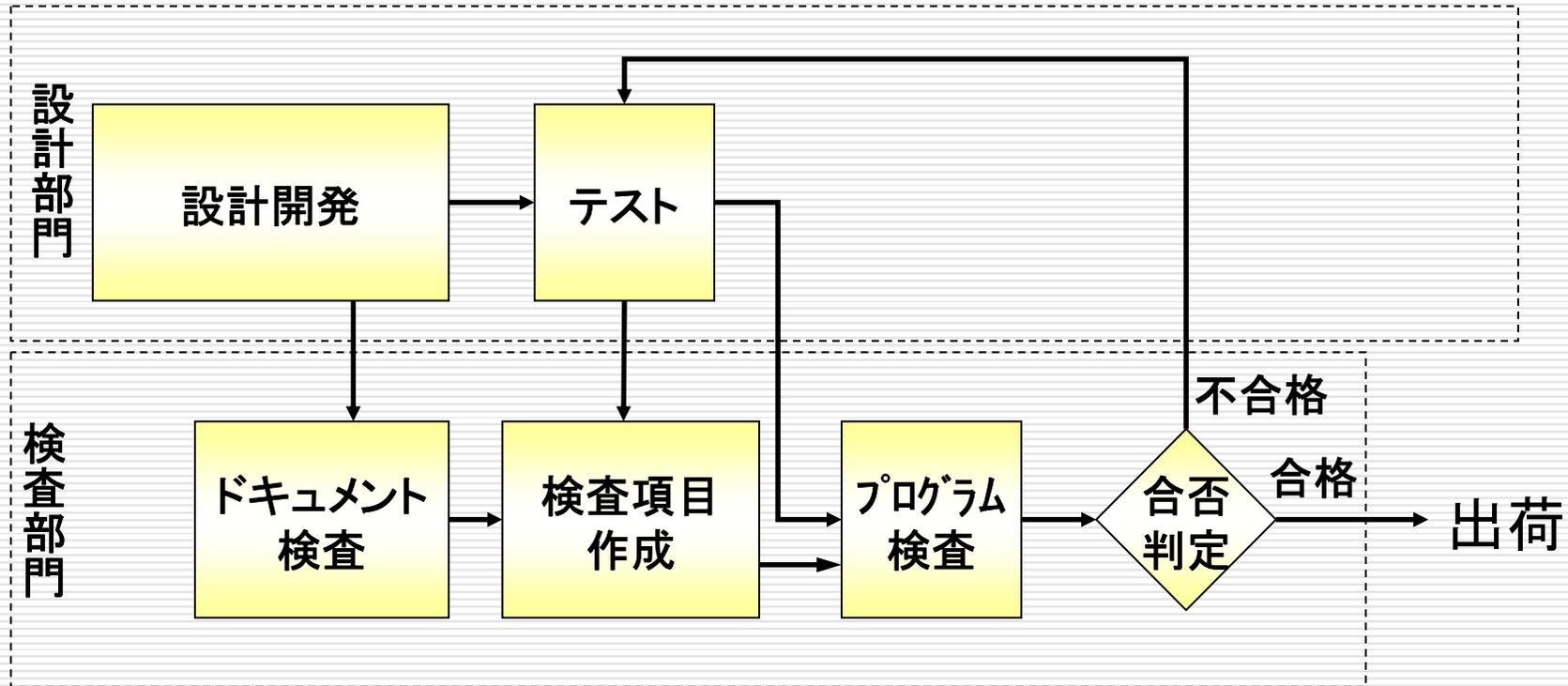
⇒物は出来たが殆どは「納期遅れ」であった

⇒この時代の成果:ドキュメント検査(中間成果物の検査、V&Vに相当)

この時代の開発エンジニアはエキサイティングであった

⇒現在の開発エンジニアはどうか?

ソフトウェア工場制度における検査（70年代）



設計工程と検査工程の関係

始まりは幹部の一言、ソフトウェアは製品⇒出荷前に検査せよ

‘70年代 技術の変化 (出展:日経システム2009, 2)

□ MIS (Management information system) の時代

■ システム構築技術

階層型/ネットワーク型 データベース

■ 開発言語

COBOL、PL/I、BASIC

■ インフラ

メインフレーム

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、QC 7つ道具、全員参加、小集団活動)
- 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 ☆☆☆ ———

QC7つ道具

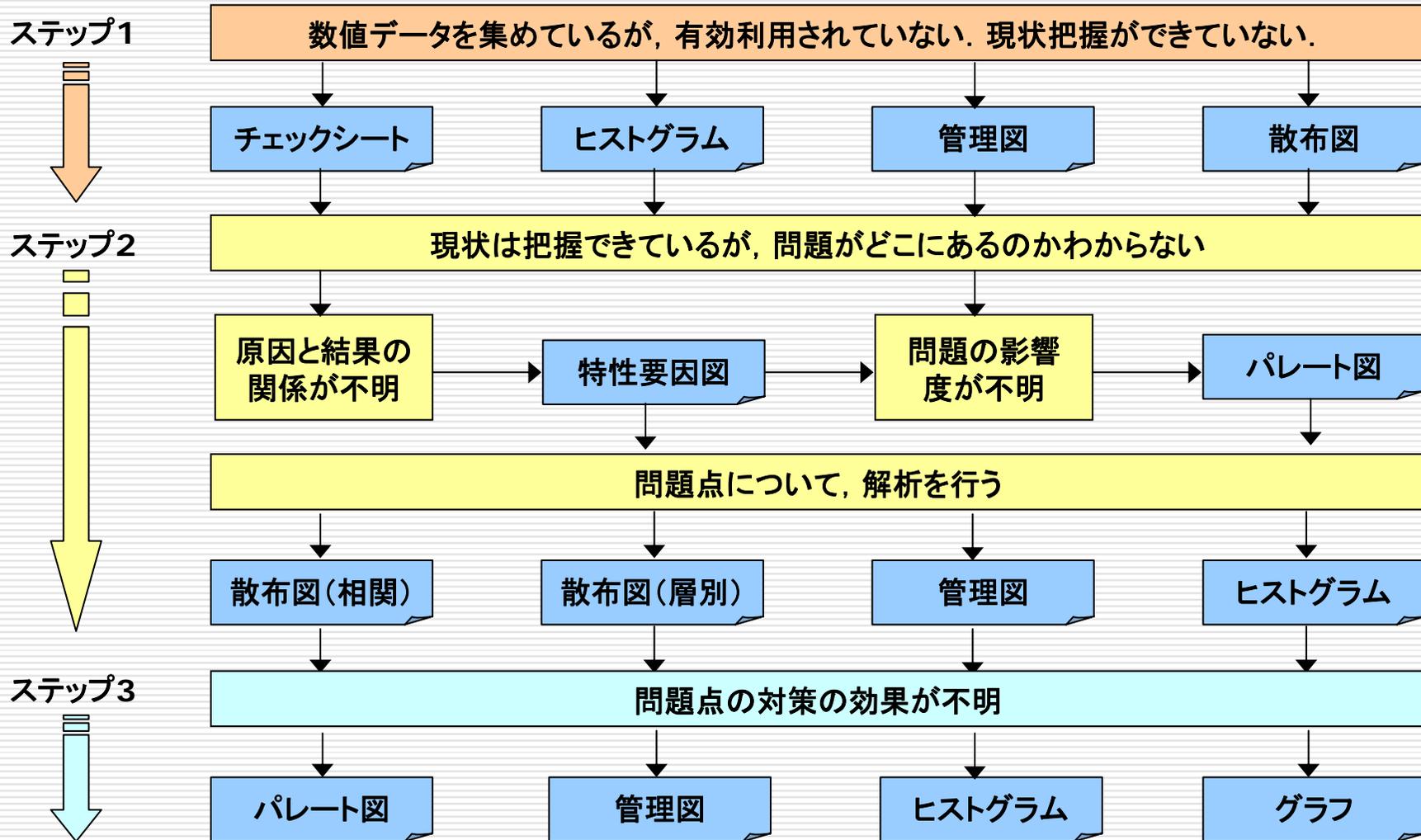
#	項目	内容
1	特性要因図	品質の特性と要因の関係を表す。普通、魚の骨に類似した形状となる。
2	パレート図	項目を横軸とし、度数の多い項目から順に度数を縦軸にとり、累積相対度数曲線を併用した図。
3	チェックシート	データの分類項目別分布を知るために使う。要因の系統的整理に際して、効率よくデータを取るのに有効。
4	ヒストグラム	データのばらつきの姿を知る。
5	散布図	2変数を横軸と縦軸にとり、値を打点して作られる図。相関性など、二つの対になったデータの検討に使う。
6	管理図	工程が安全な状態にあるかどうかを調べるため、または工程を安定な状態に保持するために持ちこたえる図。
7	グラ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

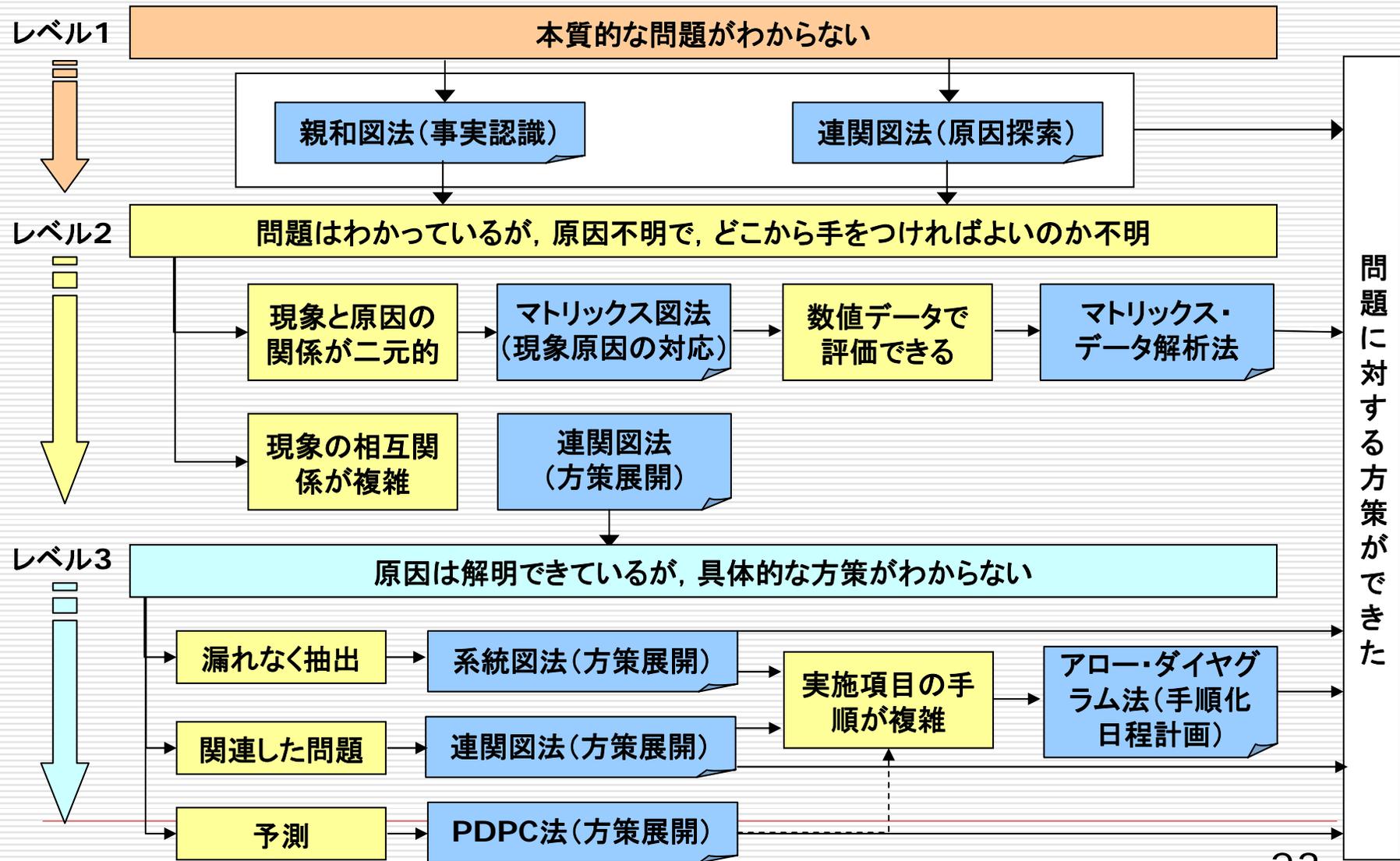
新QC7つ道具

#	項目	内容
1	系統図法	価値工学(VE)の機能分析における機能系統図を導入したもの。目的を多段の目的、手段の系列に展開する。
2	マトリックス図法	列方向と行方向に層別して、特性の関連性を明確にするための手法。目的と手段の関連性を知る。
3	マトリックス・データ解析法	主成分分析法、マトリックス図法で示されたデータ、または評価から集約した結論を得るのに使う。
4	PDPC法 (重大事故予測法)	過程決定計画図(PDPC)を応用するもの。
5	連関図法	相関の関連性を図表化し、そのつながりを明確にする過程で問題を把握する方法。論理的にとらえる。
6	親和図法 (KJ法)	問題解決技法、問題発見型、現象を数量化せず、データを言葉や文字でとらえる。総論的にとらえ、異質の統合を図る。
7	アローダイアログ法	PERTで使用する矢印図。工程管理などの実行計画に適用。主として確定事象系列の計画初期における効率化

QC七つ道具の使い分け



新QC七つ道具の使い分け



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。

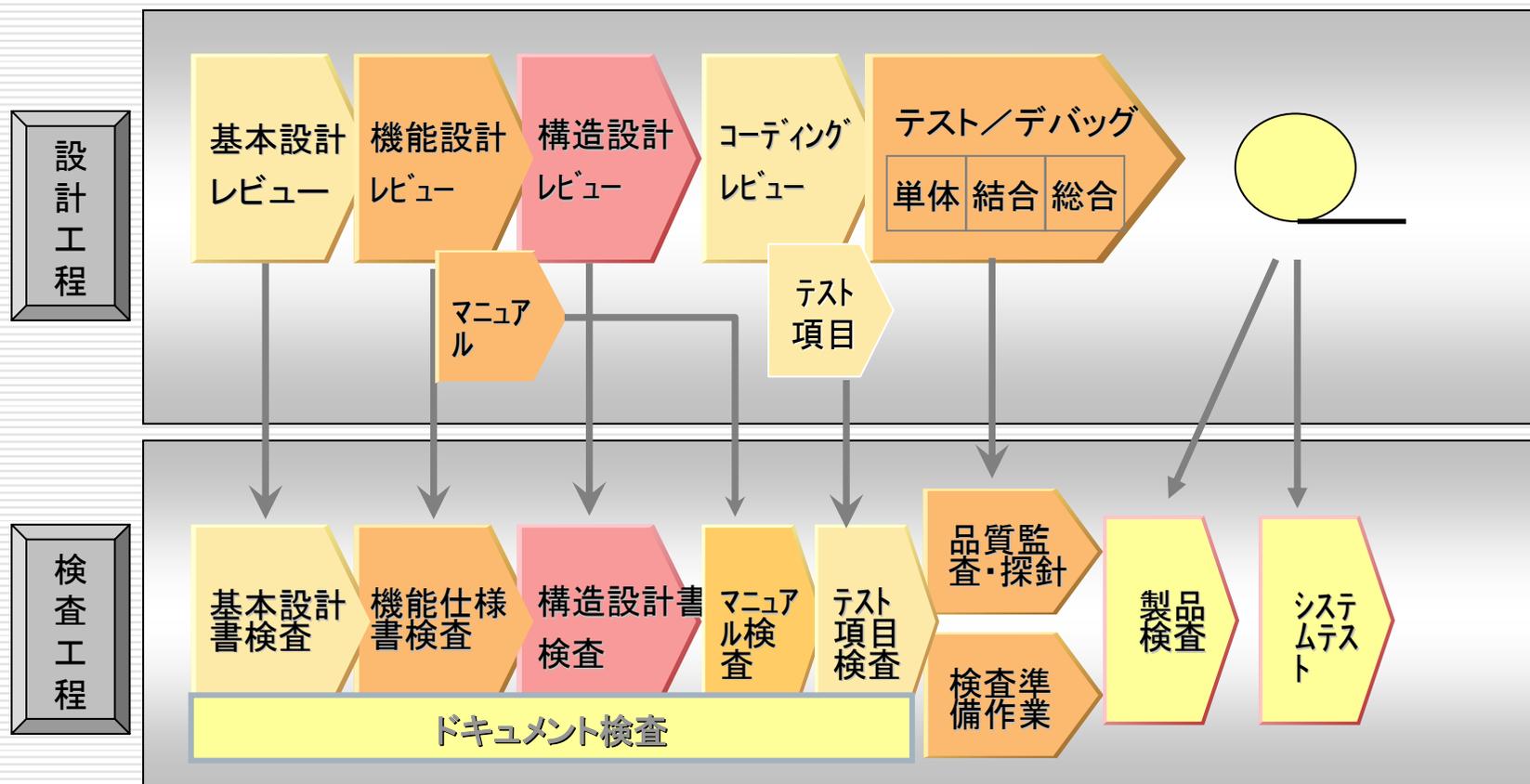
この活動を企業・組織で行なうことで...

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)ブレイン・ダイナミックス
<http://www.brain-d.co.jp/>

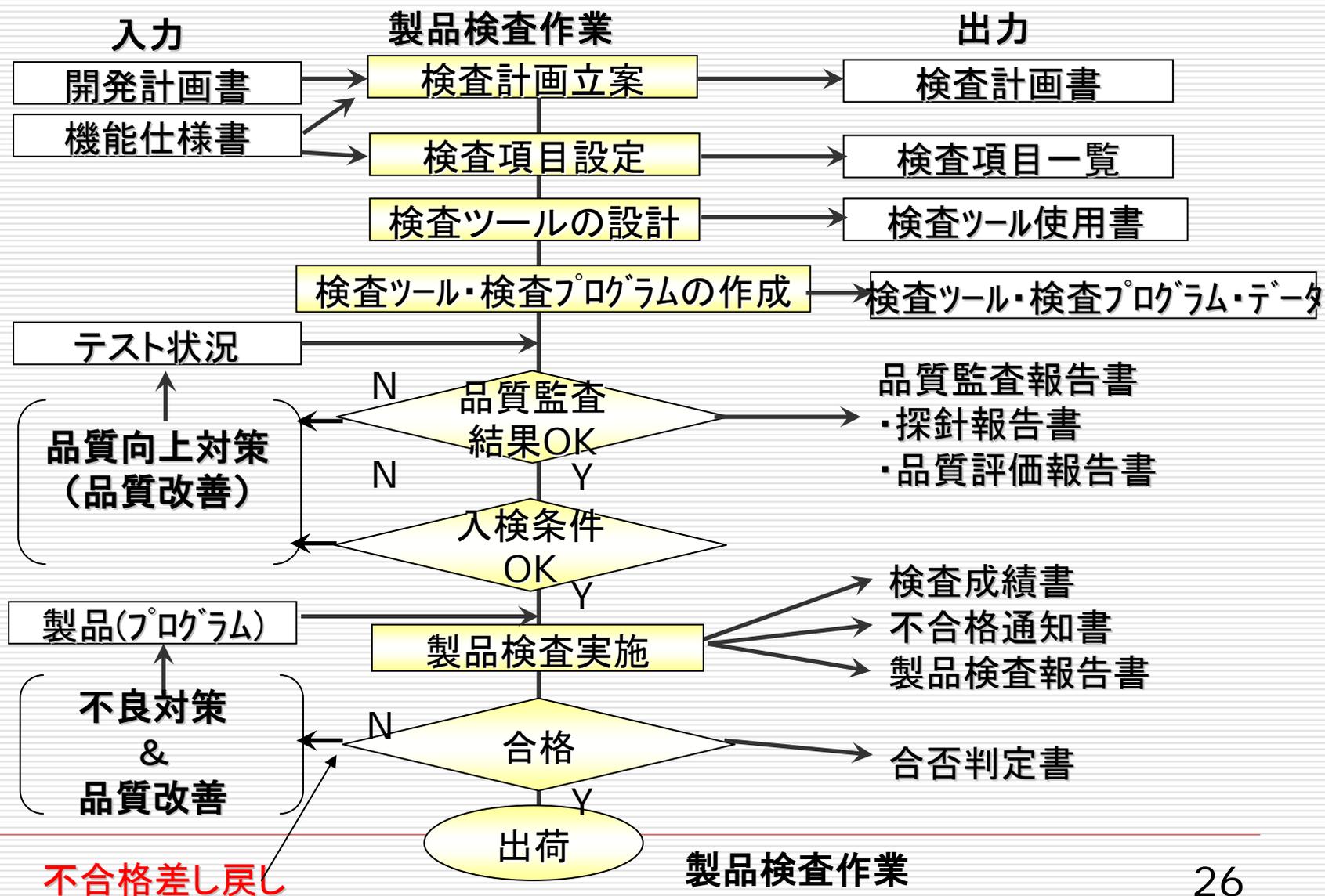
開発工程とテスト／検査の関係(80年代以降)

(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

ソフトウェア製品検査(80年代に確立)



1980年代品質保証活動②

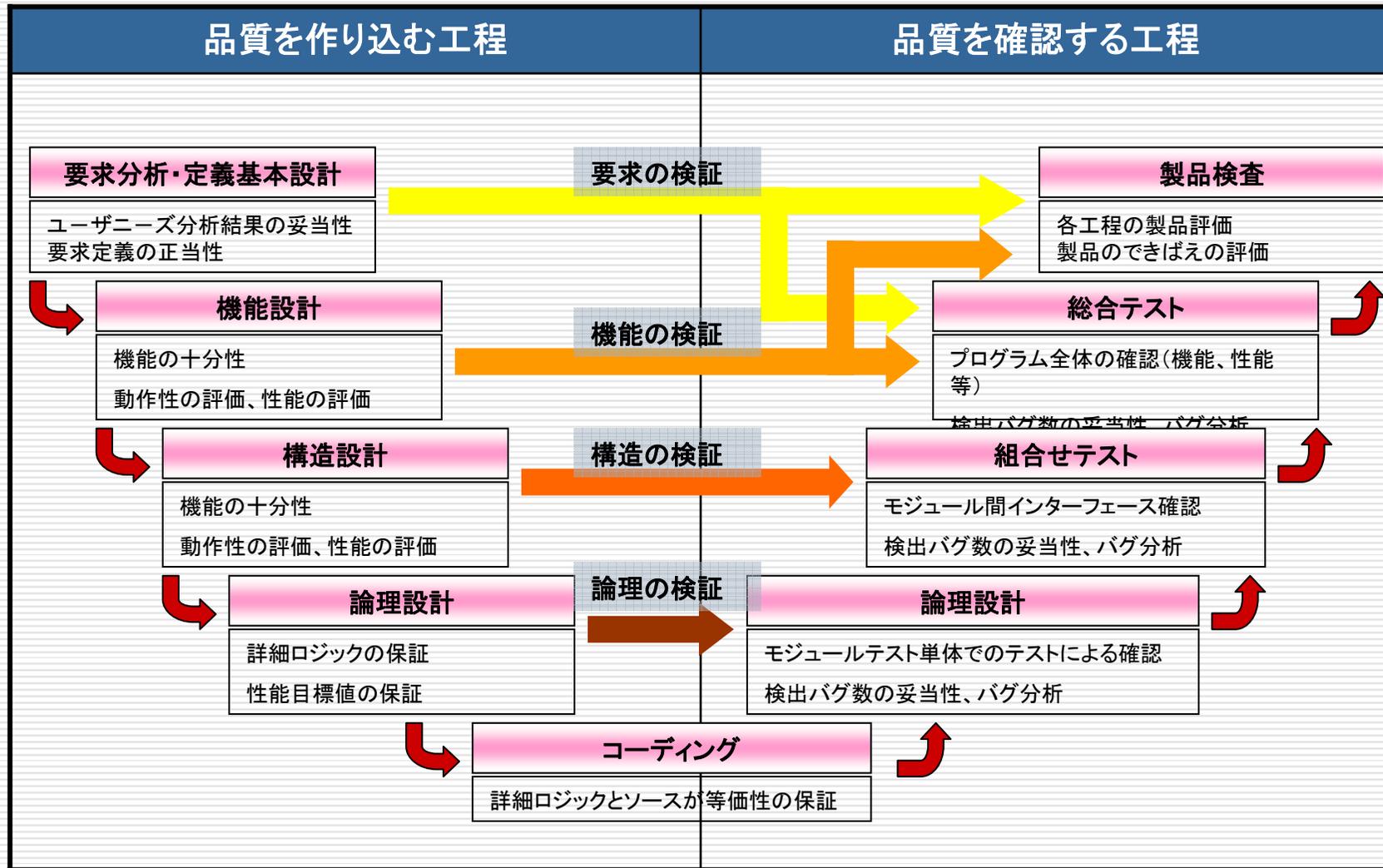
- ソフトウェア品質管理の体系化と普及
 - V字モデルに基づく管理体系の構築、普及

 - 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

<<メトリックスの必要性と重要性>>

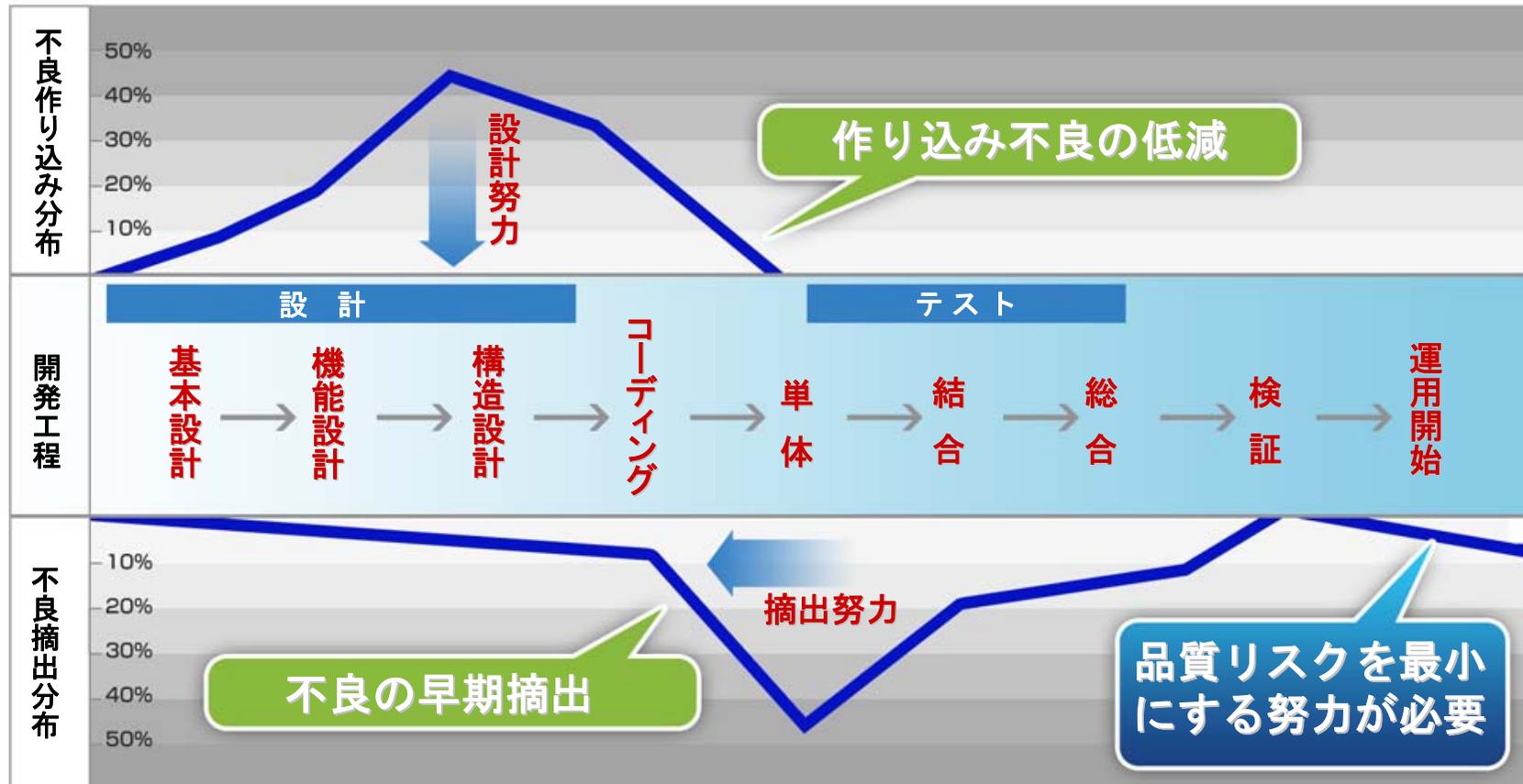
計測できないものは制御できない→トムデマルコ

V字モデル

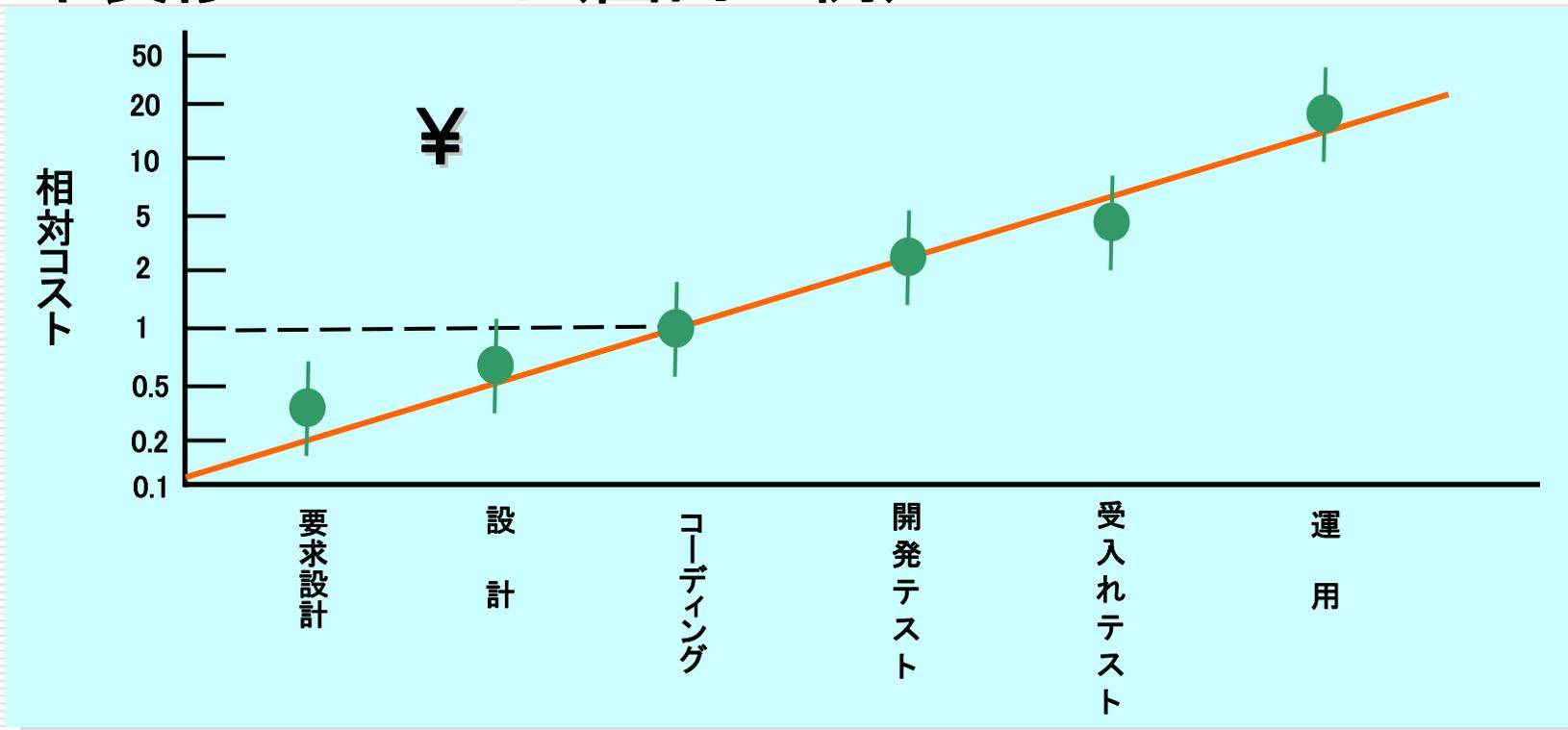


V字モデル

不良の作込みと摘出の関係



不良修正のコスト(国内の例)



- コーディング工程で作り込んだ不良
 - ➡ **テスト**で発見 : 相対コストは、約2倍
 - ➡ **運用テスト**で発見: 相対コストは、約10~50倍
- 一番コストがかかるのは、下流のテスト工程での不良発覚 (社外事故の場合、コストはMaxとなる。)

不良修正のコスト(国外の例)

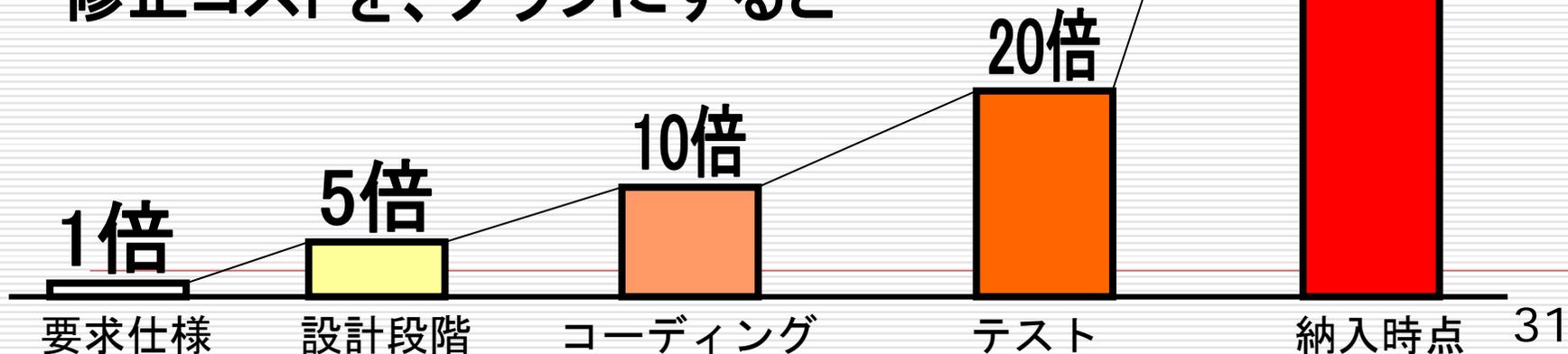
「ソフトウェア開発 201の鉄則(日経BP社)」では、原理41で要求仕様誤りの修正コストについて述べられている。

要求仕様誤りを「1」とした場合の修正コスト

設計段階.....	5倍
コーディング時...	10倍
テスト段階.....	20倍
納入時点.....	200倍

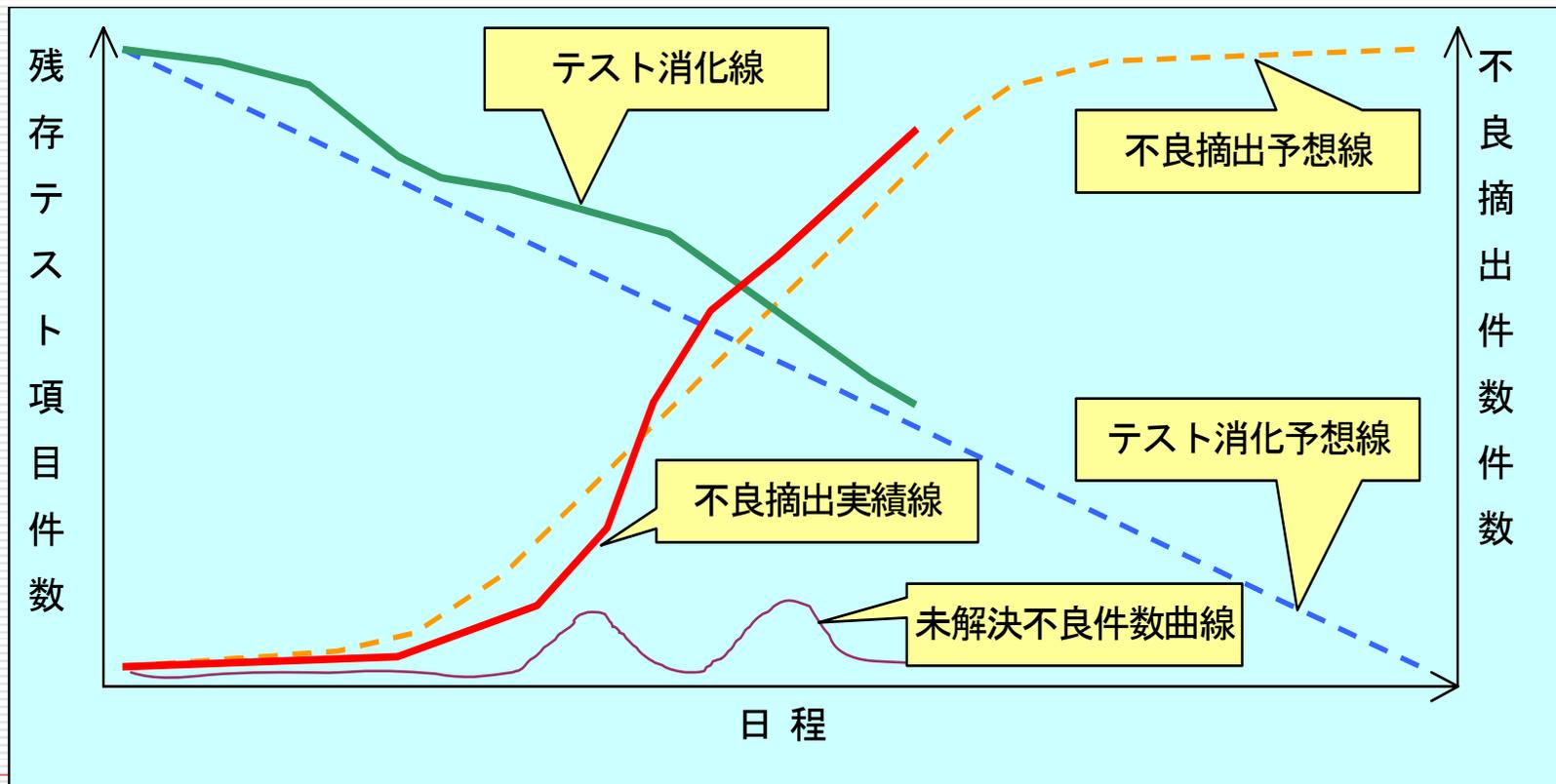
※修正コスト以外に、日程遅延や信用失墜などの損失あり

修正コストを、グラフにすると...



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

テストプロセス メトリックス

項番	工程	品質指標	単位	管理レベル	備考
1	単体 テスト	<ol style="list-style-type: none"> 不良摘出件数（累積） 不良摘出件数/KS テスト項目進捗度 不良摘出件数/投入工数 不良摘出件数/マシン時間 テストプログラム消化率 不良率（不良摘出件数/テスト項目数） 	件 件/KS % 件/人H % % %	プログラム、モジュール単位 モジュール単位 モジュール単位 モジュール単位 モジュール単位 モジュール単位 モジュール単位	成長曲線
2	プログラ ムテ スト	<ol style="list-style-type: none"> 不良摘出件数（累積） 不良摘出件数/KS テスト項目進捗度 不良摘出件数/投入工数 不良摘出件数/マシン時間 テストプログラム消化率 設計総合耐久テストの故障率 探針での不良率 探針での故障率 探針での推定残不良数 不良率（不良摘出件数/テスト項目数） 	件 件/KS % 件/人H % % 件/H % 件/H 件 %	プログラム、モジュール単位 モジュール単位 モジュール単位 モジュール単位 モジュール単位 モジュール単位 システム、プログラム単位 プログラム単位 プログラム単位 プログラム単位	成長曲線 (デバッグ効率)
3	製品 検査	<ol style="list-style-type: none"> 不合格件数 不合格件数/KS 製品検査実施回数 故障率（1/MTBF） MTBD 不良率 不良摘出件数（累積） 	件 件/KS 回 件/H H/件 % 件	システム、プログラム単位 プログラム単位 プログラム単位 プログラム単位 プログラム単位 プログラム単位 プログラム単位	(設計発見不良)

テストプロセスでの信頼性評価尺度

項番	品質特性	適用工程			計測尺度	定義
		開発	検査	稼働		
1	成熟性	○	○	○	障害密度	障害件数／規模(KS)
2		○	○		障害率	障害件数／テスト項目数
3		○	○	○	故障率	故障件数／マシン時間
4		○	○	○	平均故障発生間隔	稼働時間／故障件数
5		○	○	○	平均ダウン発生間隔	マシン時間／ダウン件数
6		○	○		障害収束率	累積摘出障害数／推定障害総数
7				○	探針不良率	探針不良件数／探針実施項目数
8		○	○		現象別障害比率	現象別障害件数／全摘出障害件数
9		○	○		原因別障害比率	原因別障害件数／全摘出障害件数
10	テスト品質	○	○		テスト密度	テスト項目数／規模(KS)
11		○	○		テスト実施率	実施済みテスト項目数／実施予定テスト項目数
12		○			テストカバレッジ率(C0)	実施済みステートメント数／全ステートメント数
13		○			テストカバレッジ率(C1)	実施済み分岐数／全分岐数
14	可用性		○	○	稼働率	実稼働総時間／予定稼働時間
15			○	○	平均復旧時間	復旧に要した総時間／ダウン回数
16				○	重症不良件数頻度	件／月

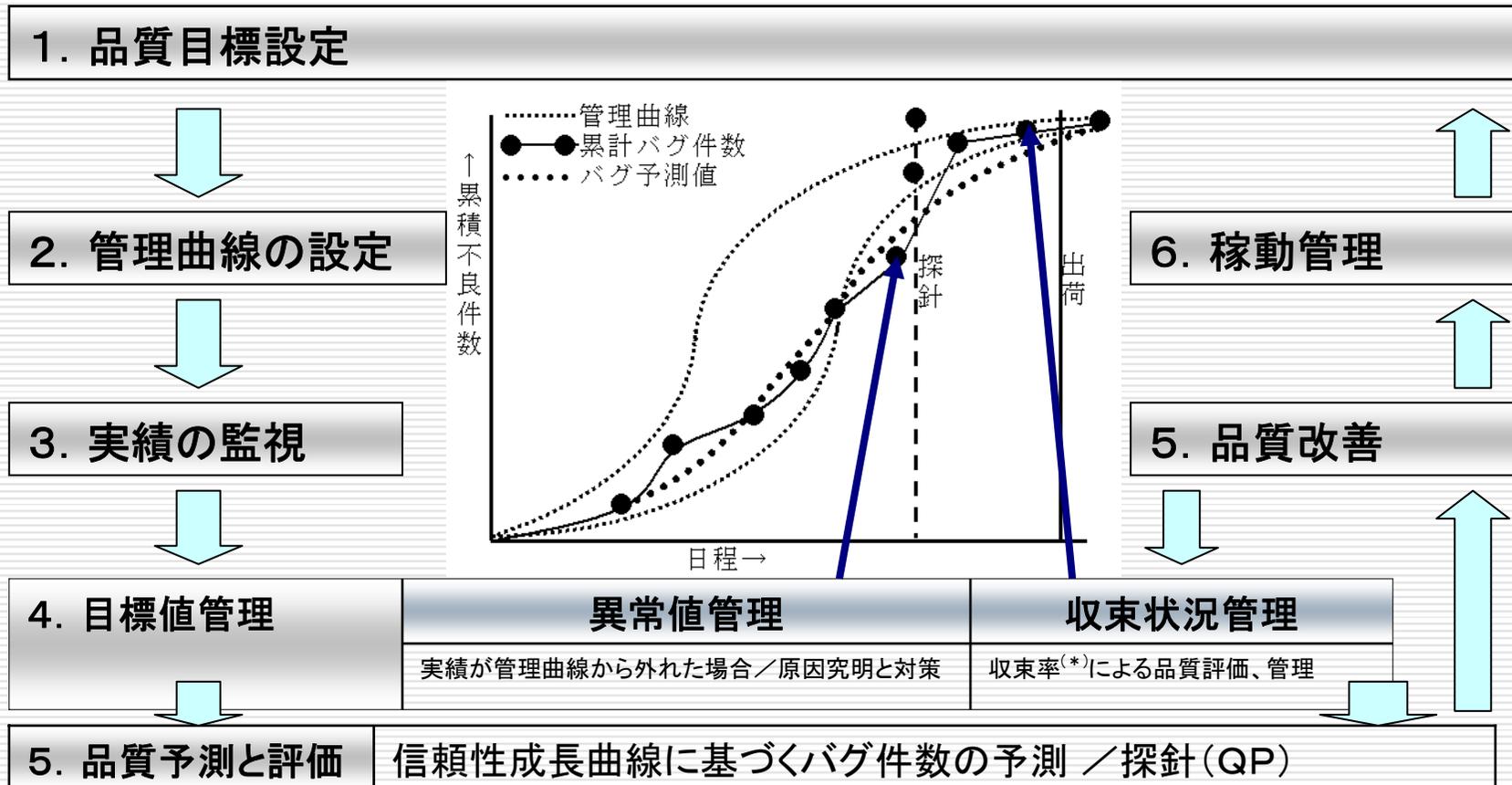
目標値設定(テスト工程)の具体例

バグ抽出／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
<p>1.PCL件数: 100件/kloc</p> <p>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</p> <p>3.バグ抽出件数: 8~10件/kloc</p>	<p>1.CCL件数: 20~25件/kloc</p> <p>2.バグ抽出件数: 1~2件/kloc</p>	<p>1.SCL件数: 5~10件/kloc</p> <p>2.バグ抽出件数: 0.5件/kloc</p>

* PCL : 単体テスト項目、CCL : 結合テスト項目、SCL : 総合テスト項目

品質目標値管理の概念図



※収束率 = 実績バグ件数 / 予測バグ件数

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。

$$\text{上限値: } P_U = \rho' + u(\alpha) \sqrt{\rho'(1-\rho')/n} \quad u(\alpha)=1.9$$

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

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

信頼性成長モデルの具体例

□ 時間計測モデル

ソフトウェア故障発生時間あるいはエラー発見時間に基づく確率・統計モデル。

□ 個数計測モデル

発生したソフトウェア故障数、あるいは発見されたエラー数に基づく確率・統計モデル。代表的なものに、NHPPモデルや超幾何分布モデルがある。

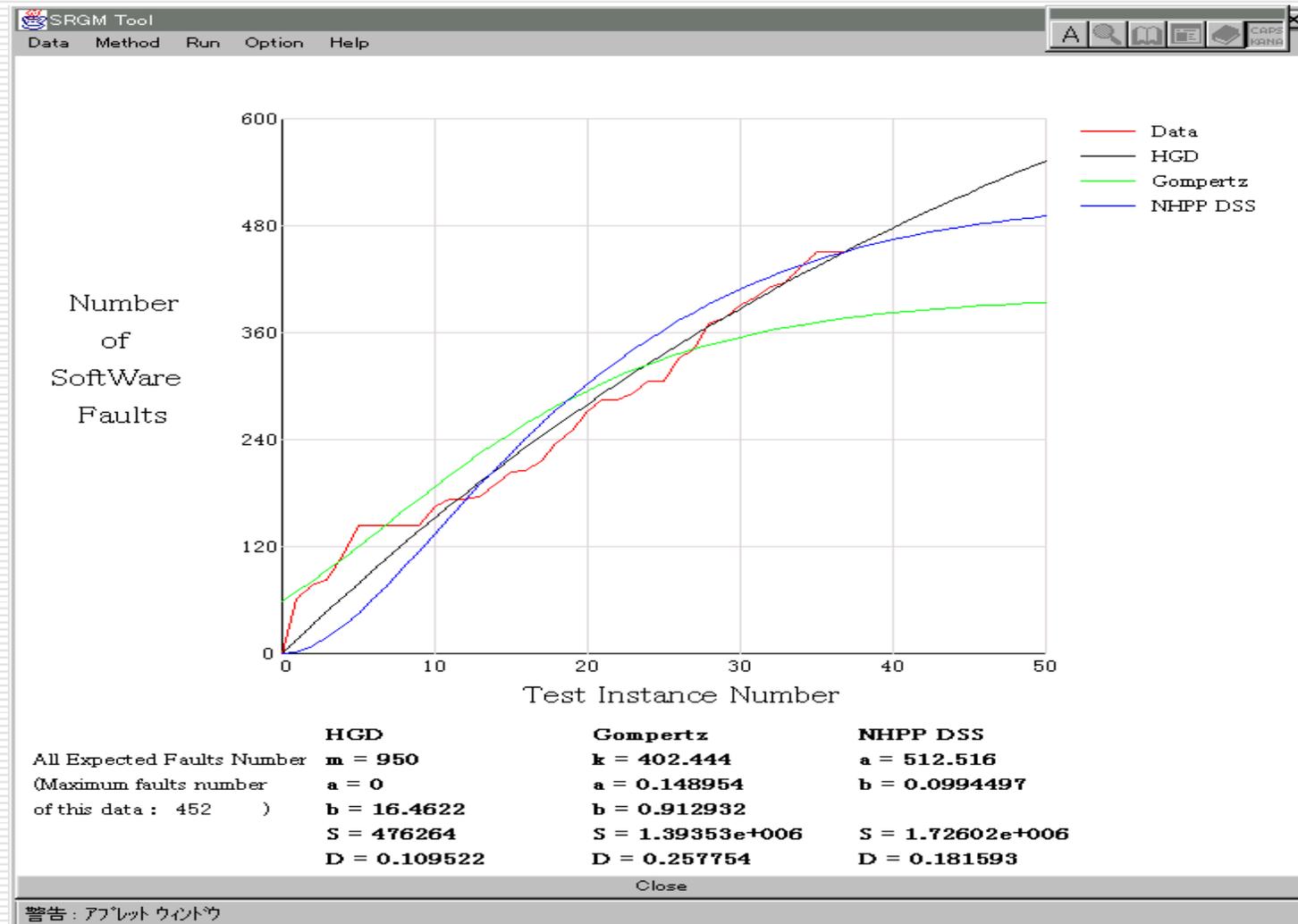
□ アベイラビリティモデル

ソフトウェアの時間的挙動を、ソフトウェア故障の発生しない動作状態と、ソフトウェア故障の発生した不動作状態により記述する確率モデル。

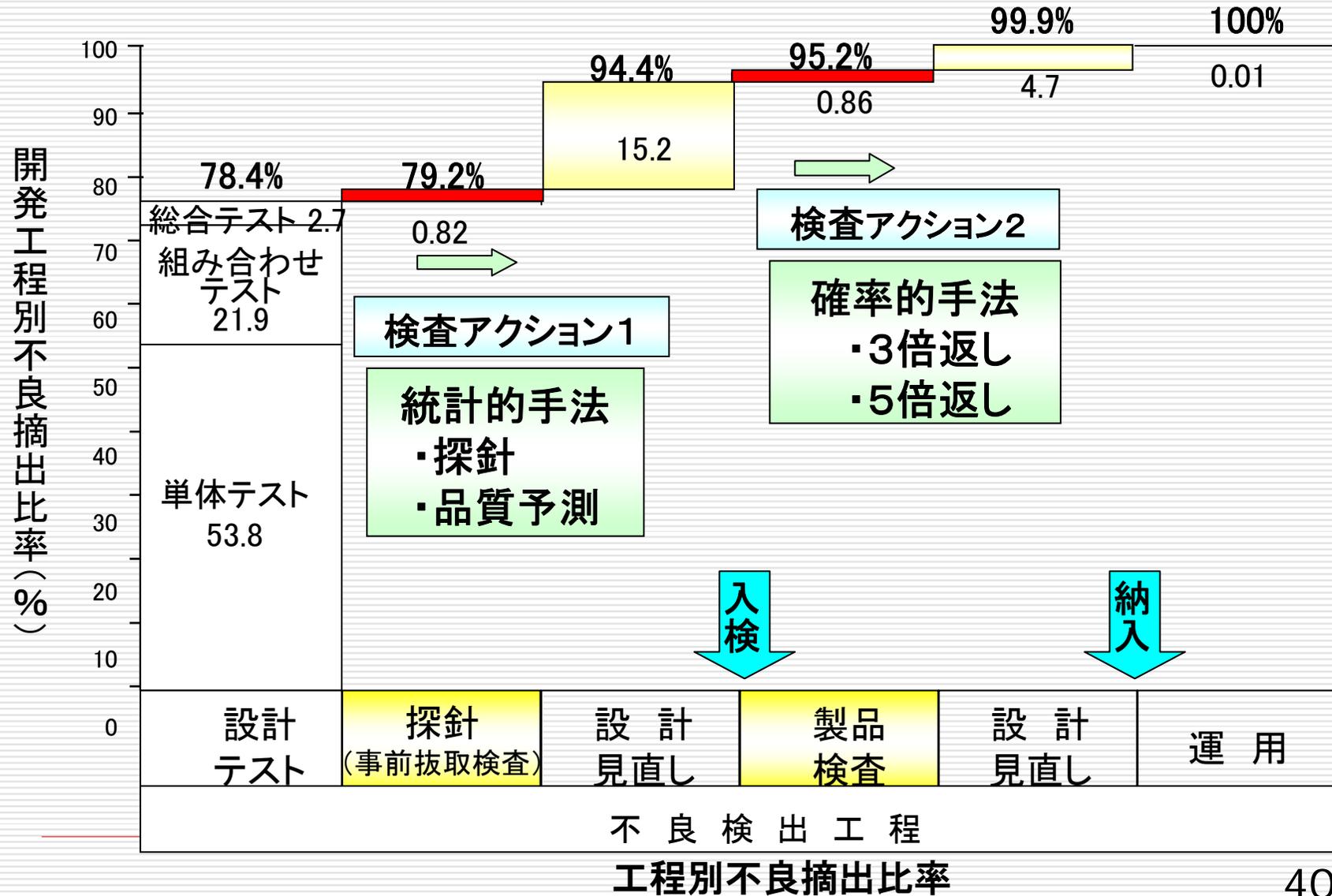
□ 傾向曲線モデル

累積バグ発生数がS字形成長曲線に類似しているという事実を利用し、バグの総出現数や、デバッグの完了時期等を推定する。ゴンベルツ曲線モデルや、ロジスティック曲線モデルがある。

信頼性成長モデルの具体例



開発工程と不良摘出



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 * *

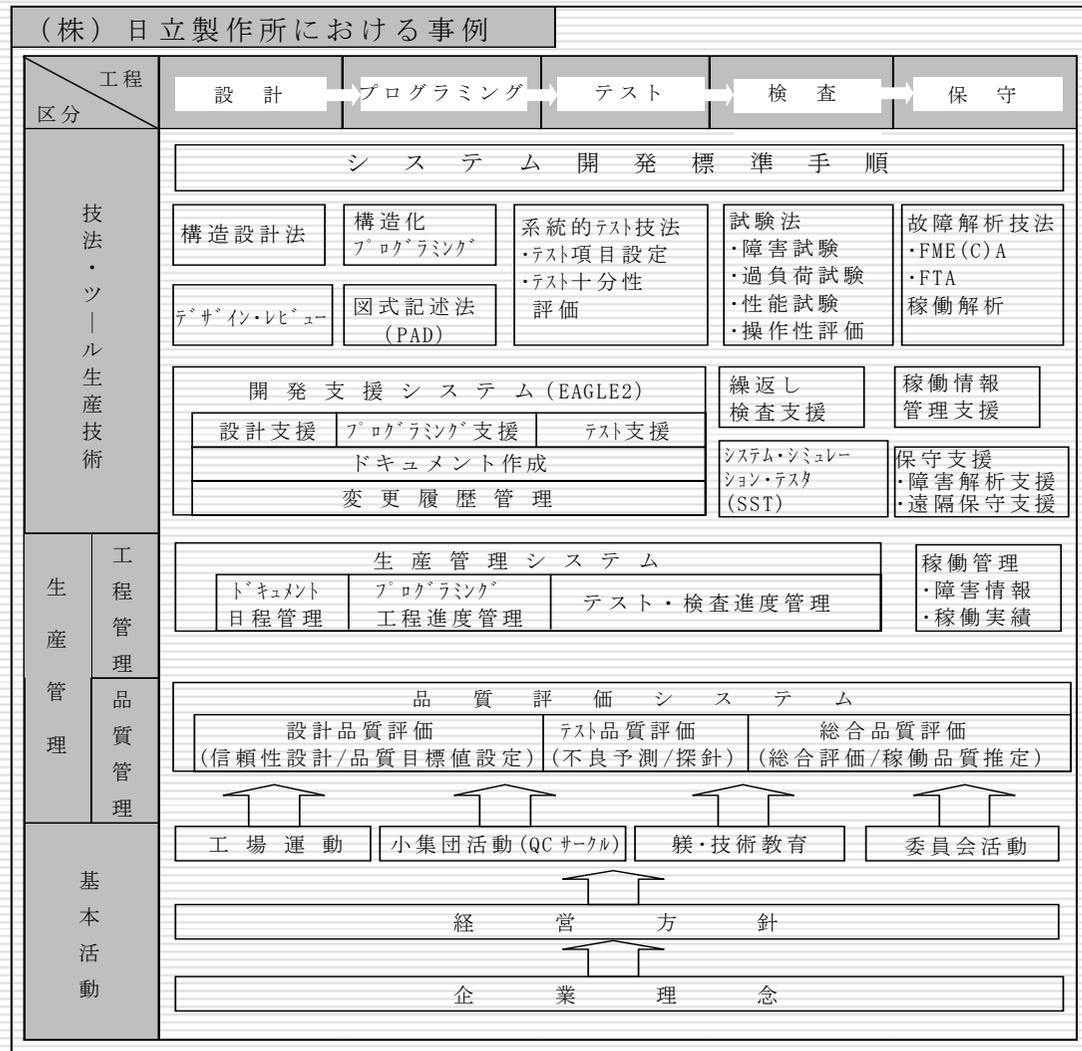
* * 品質管理の隆盛期、現在の手法はこの時代に
確立された * *

* * **これが現在に継承され、進化しているか(?)** * *

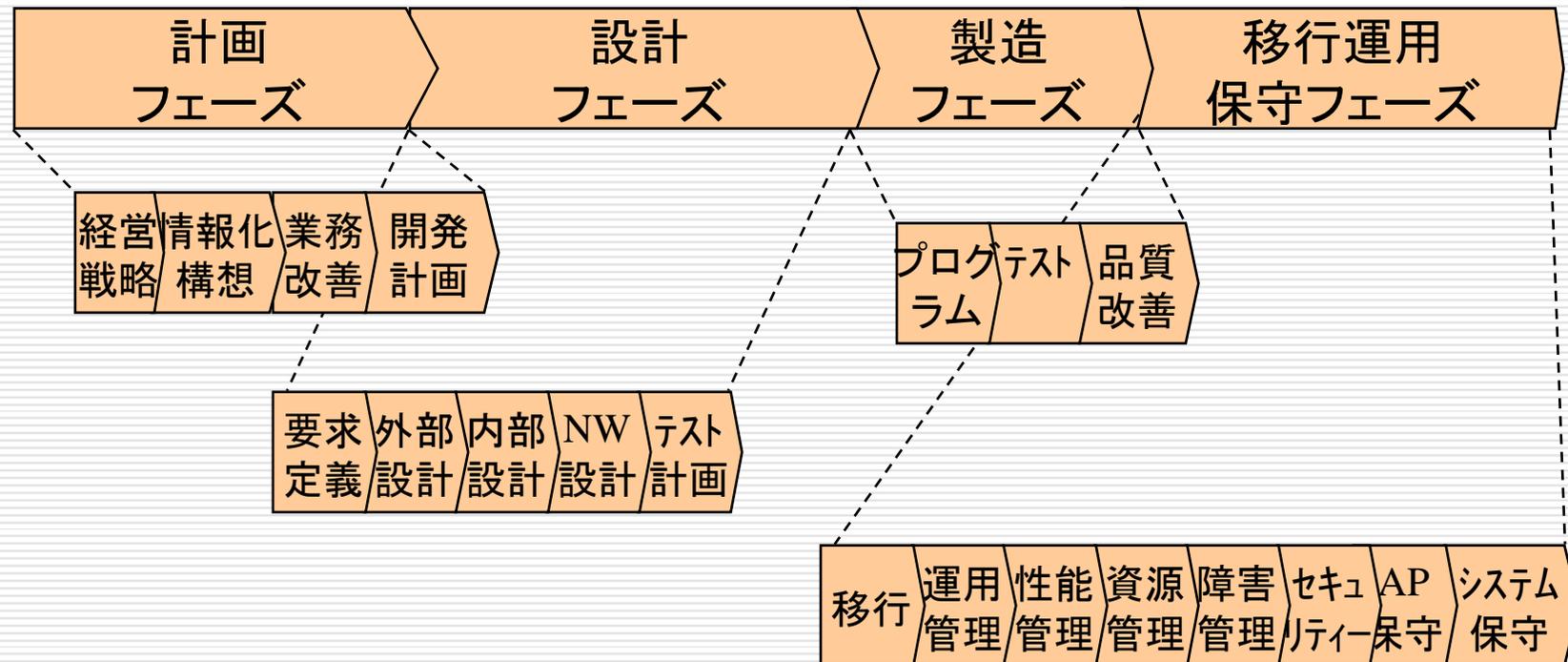
1980年代品質保証活動④

- ソフトウェア開発の体系構築
 - 70年代の経験を開発標準として定義、体系化
 - 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

ソフトウェア開発／品質保証の体系(標準)化

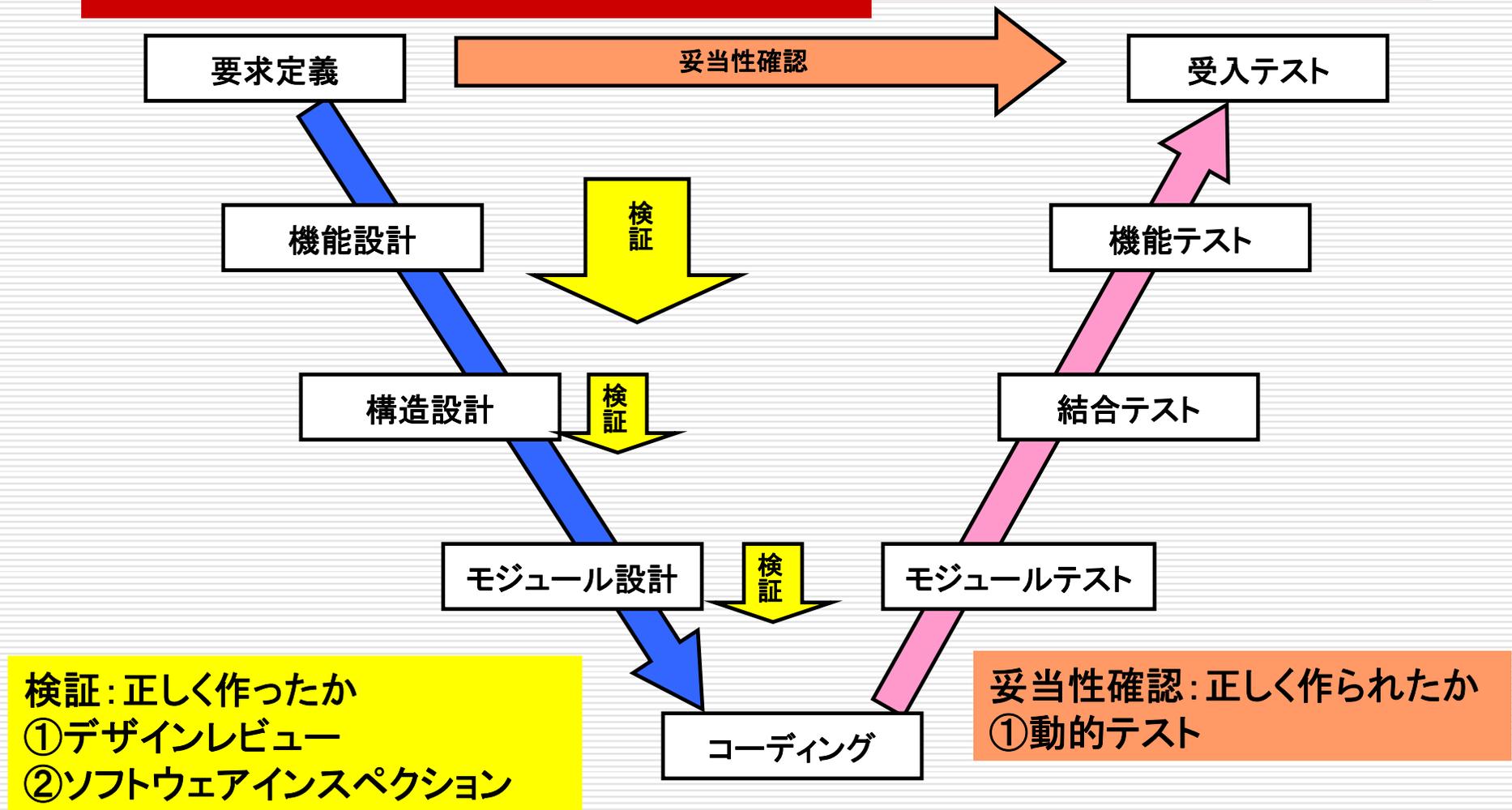


ソフトウェア開発プロセスの定義



- 開発方法論の基本はウォーターフォールモデル
⇒ 80年代に基本形確立、以降徐々に詳細化が進んだ

V & V (検証と妥当性の確認)



•「検証」と「妥当性確認」とは？

• 検証 (Verification)

- 「検証」の目的は、選択された作業成果物が、指定された要件を満たすことを確実にすることである
 - 「それを正しく構築した」ことを保証する **making the thing right**

• 妥当性確認 (Validation)

- 「妥当性確認」の目的は、成果物または成果物構成要素が、意図した環境に設置されたときにその使用意図を充足することを実証することである
 - 「正しいものを構築した」ことを保証する **making the right thing**

•「検証」と「妥当性確認」とは？—参考

- IV & Vと呼ばれることも

- 検証の独立性を重視する観点から、[IV&V](#)
(Independent Verification and Validation: 独立検証と妥当性確認)と呼ばれることもある

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.)がレビューを行う

- 色々なレビュー

- ・ウォークスルー
- ・ピアレビュー
- ・コードレビュー

レビューの定義と概要

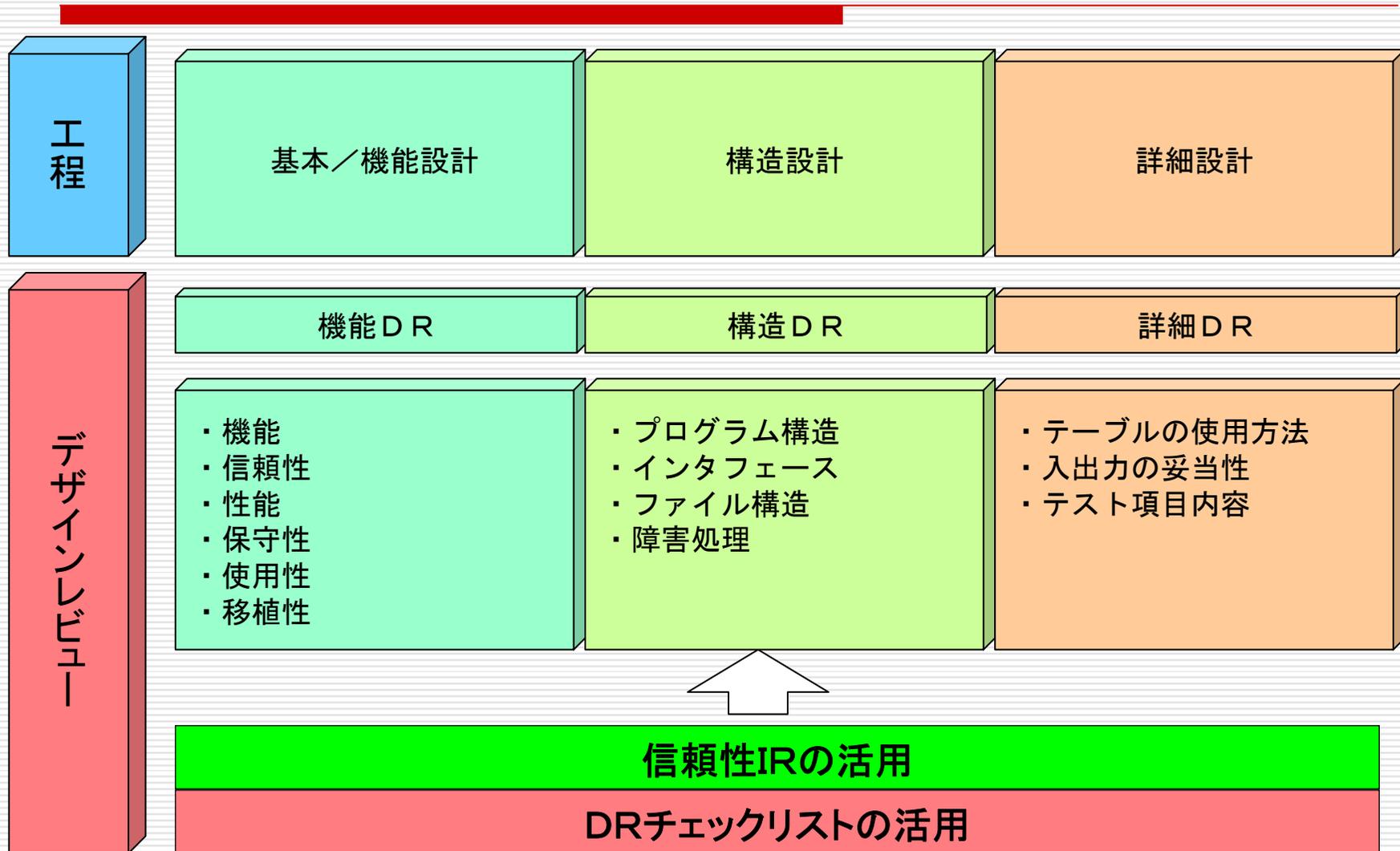
(1) デザインレビューとは

- 設計工程の主要な切れ目で行われる見直し作業
- 審査で設計上の欠陥を摘出、修正と改善余地の検討

(2) デザインレビューの目的

- ① 成果物の品質評価(不良摘出を含む)による品質向上
- ② 成果物の評価による進捗状況の把握
- ③ 上流工程での品質作り込みによる、手戻り削減と生産性向上
- ④ レビュー結果を開発技術／技法やプロセス改善に役立てる
- ⑤ 担当者以外の仕様理解者を増やす、レビュー参加者のOJT

デザインレビューの要点

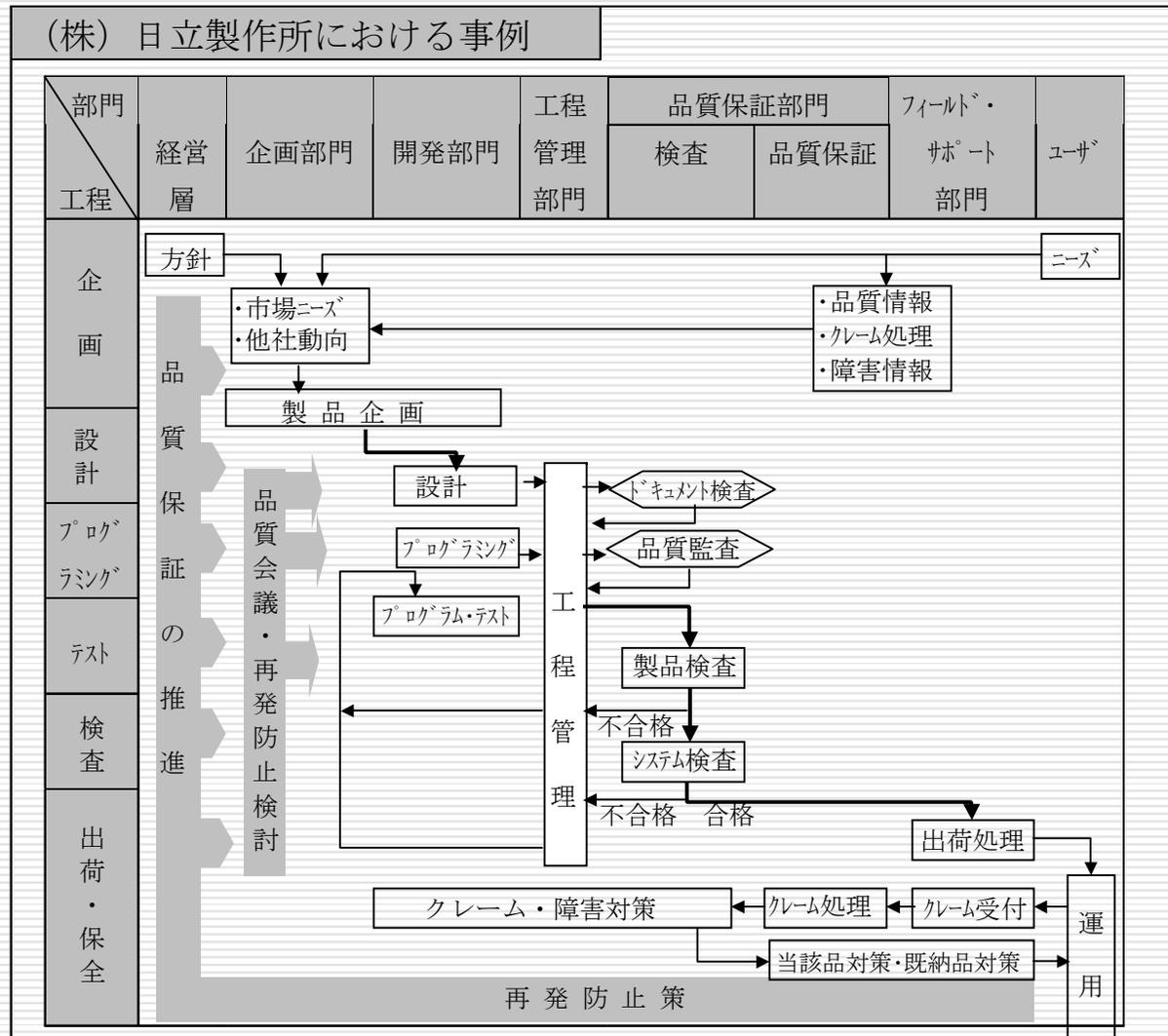


1980年代品質保証活動⑤

- ソフトウェア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

開発支援システム (EAGLE)		
設計支援	プログラミング支援	テスト支援
ドキュメント作成支援		
変更履歴管理支援		

ソフトウェア品質保証システム(支援部門の整備)



品質保証システム

‘80年代 技術の変化 (出展:日経システム2009, 2)

□ SIS(Strategic information system)の時代

■ システム構築技術 RDB, CASEツール

■ 開発言語 C/C++、 Pascal、 Smalltalk, BASIC

■ インフラ オフコン

1990年代 品質保証活動(前半)①

□ 背景: 80年代品質管理の閉塞感とISO9000の台頭

⇒ 品質管理では不十分、プロセス管理へ

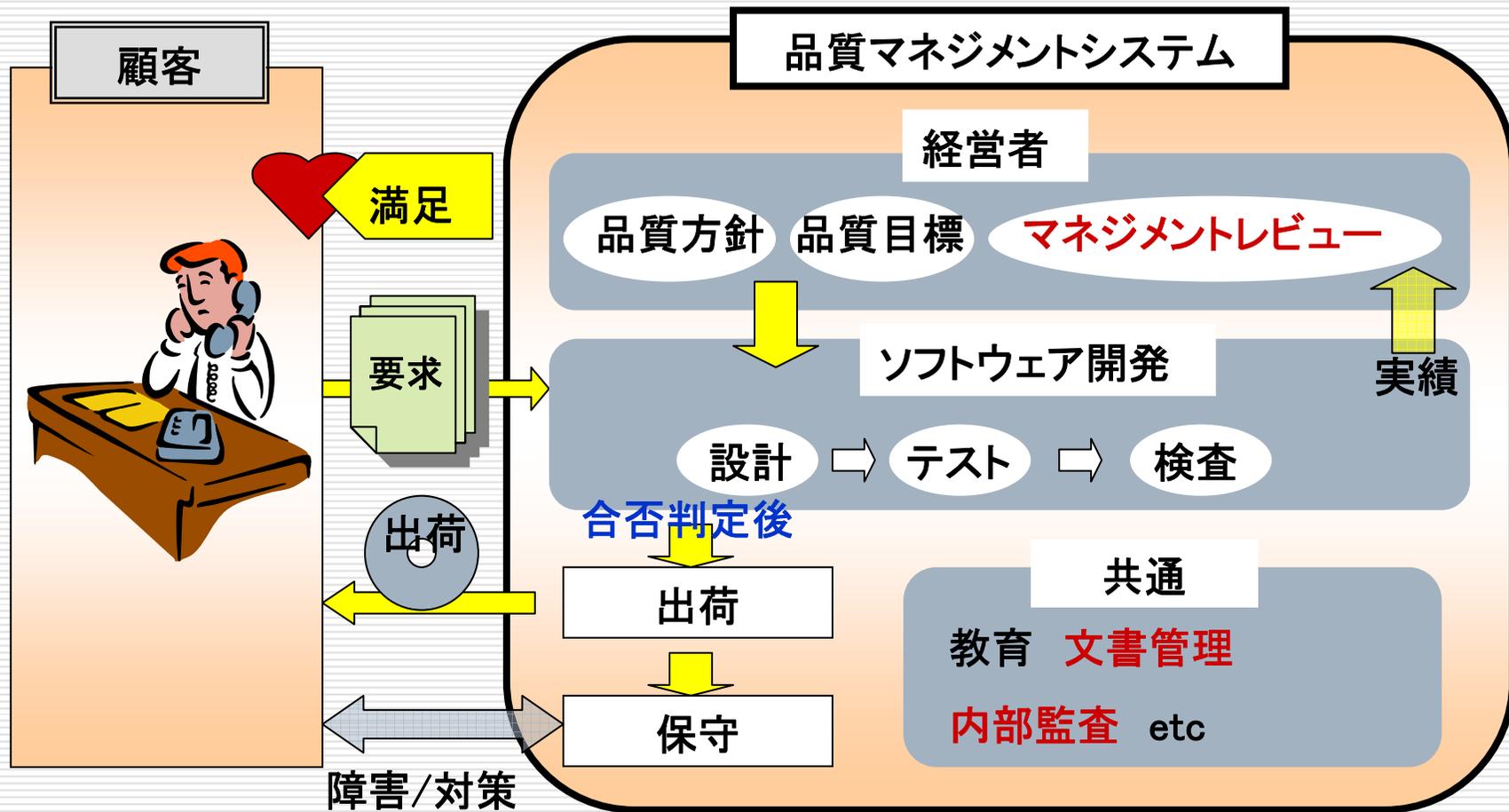
- 良いプロセスが良いプロダクトを生み出す → SPI の時代の開幕
- プロダクト重視からプロセス重視へ

□ 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 ***

ISO9000の概要

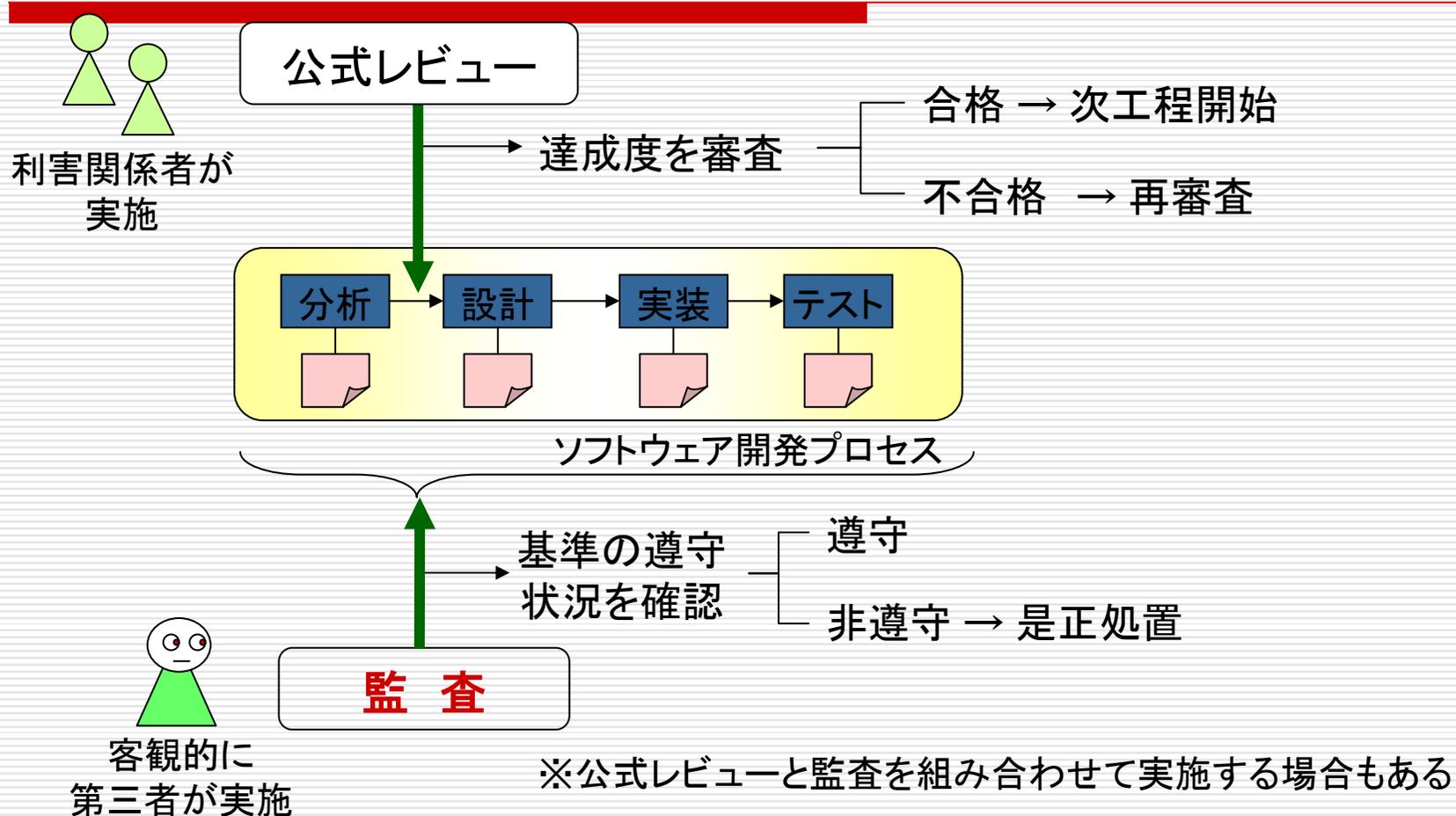


監査とは(1/2)

- 「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.)の遵守状況を報告する
 - 客観性をもって実施

監査とは(2/2)



□ 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

1990年代 品質保証活動(前半) ②

□ SW-CMM発表

■ プロセス重視に拍車→支援部門主体

■ 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 >>

1990年代(中半～)品質保証活動①

- 背景:オープン(ネオダマ)化時代の到来、ITバブルの崩壊
 - 信頼性重視→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**

(注) ネオダマ; ネットワーク、オープン、ダウンサイジング、マルチメディア

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ト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

PMBOK(PMI)のPM定義

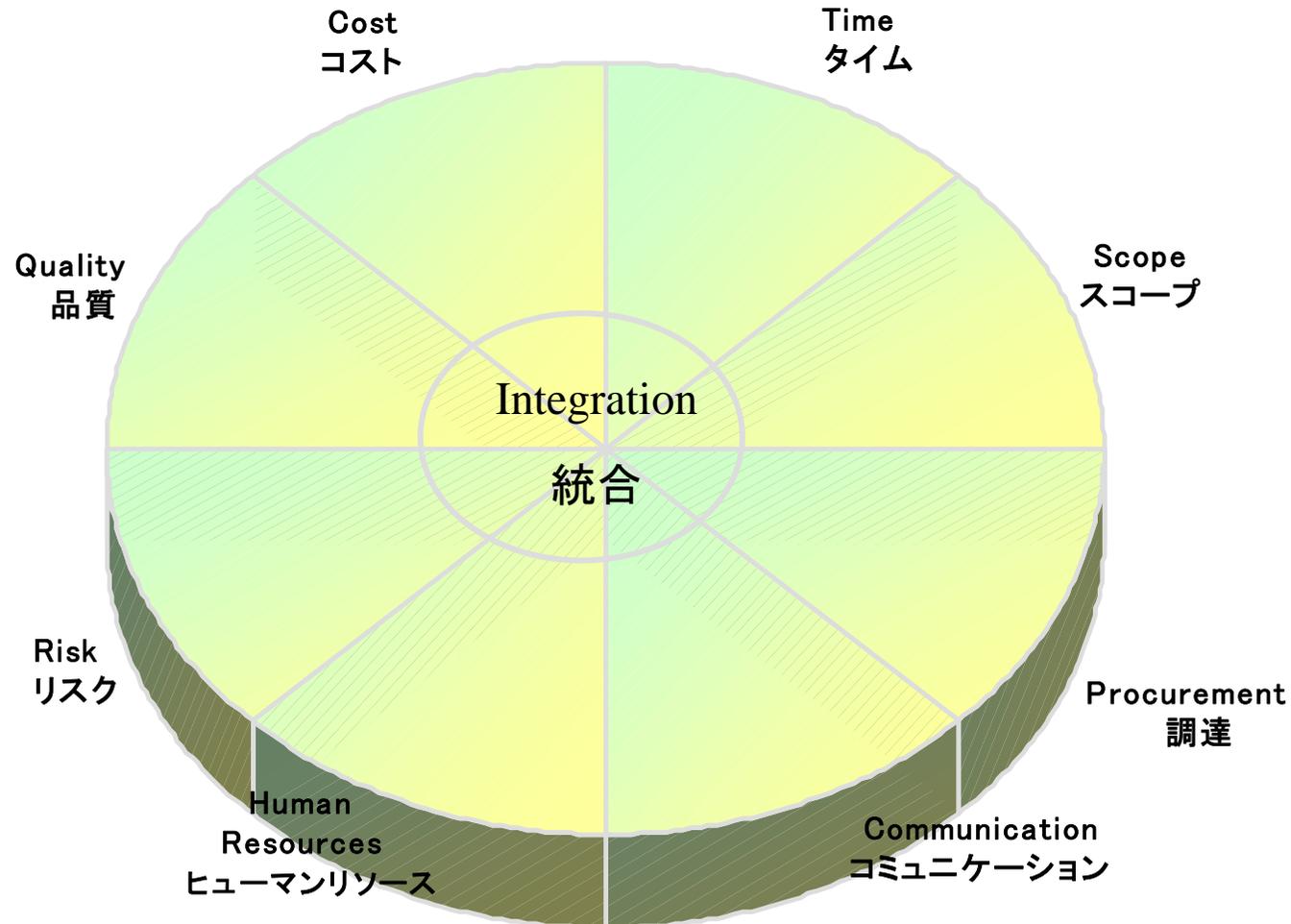
- ◆プロジェクトマネジメントとは、プロジェクトの要求事項を満足させるために、知識、スキル、ツールと技法をプロジェクト活動へ適用する事である。プロジェクトマネジメントは、立ち上げ、計画、実行、監視コントロール、終結などのプロジェクトマネジメントプロセスの適用、統合を行うことによって達成される。プロジェクトマネジャーは、プロジェクト目標を達成する責任を負う人である。

- ◆プロジェクトマネジメントには、以下の事項が含まれる。

- ・要求事項を特定すること。
- ・明確で達成可能な目標を確立すること。
- ・品質、スコープ、タイム、コストなど、競合する要求のバランスをとること。
- ・各種ステークホルダーの異なる関心と期待に対して、仕様、計画、取り組み方法を適用させること。

PMBOKの知識エリア

PMI Project management Knowledge Areas



5. プロジェクトマネジメントの知識エリア(9つの知識エリア)

プロジェクトマネジメント

統合マネジメント

1. プロジェクト計画書作成
2. プロジェクトスコープ記述書暫定版作成
3. プロジェクトマネジメント計画書作成
4. プロジェクト実行の指揮マネジメント
5. プロジェクト作業の監視コントロール
6. 統合変更管理
7. プロジェクト終結

スコープマネジメント

1. スコープ計画
2. スコープ定義
3. WBS作成
4. スコープの検証
5. スコープコントロール

タイムマネジメント

1. アクティビティ定義
2. アクティビティ順序設定
3. アクティビティ資源見積り
4. アクティビティ所用時間見積り
5. スケジュール作成
6. スケジュールコントロール

コストマネジメント

1. コスト見積もり
2. コストの予算化
3. コストコントロール

品質マネジメント

1. 品質計画
2. 品質保証
3. 品質管理

人的資源マネジメント

1. 人的資源計画
2. プロジェクトチーム編成
3. プロジェクトチーム育成
4. プロジェクトチームのマネジメント

コミュニケーションマネジメント

1. コミュニケーション計画
2. 情報の配布
3. 実績報告
4. ステークホルダーマネジメント

リスクマネジメント

1. リスクマネジメント計画
2. リスクの識別
3. 定性的リスク分析
4. 定量的リスク分析
5. リスク対応計画
6. リスクの監視コントロール

調達マネジメント

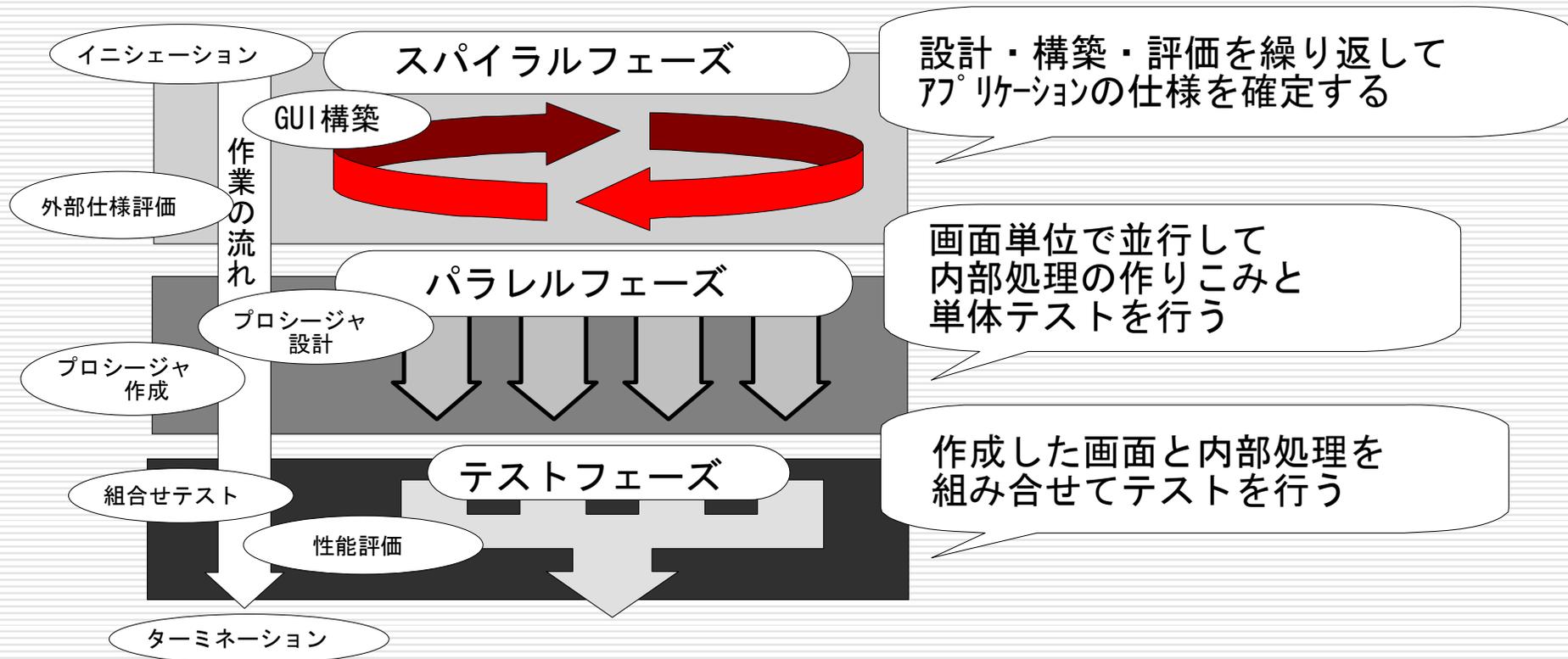
1. 購入・取得計画
2. 契約計画
3. 納入者回答依頼
4. 納入者選定
5. 契約管理
6. 契約の集結

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

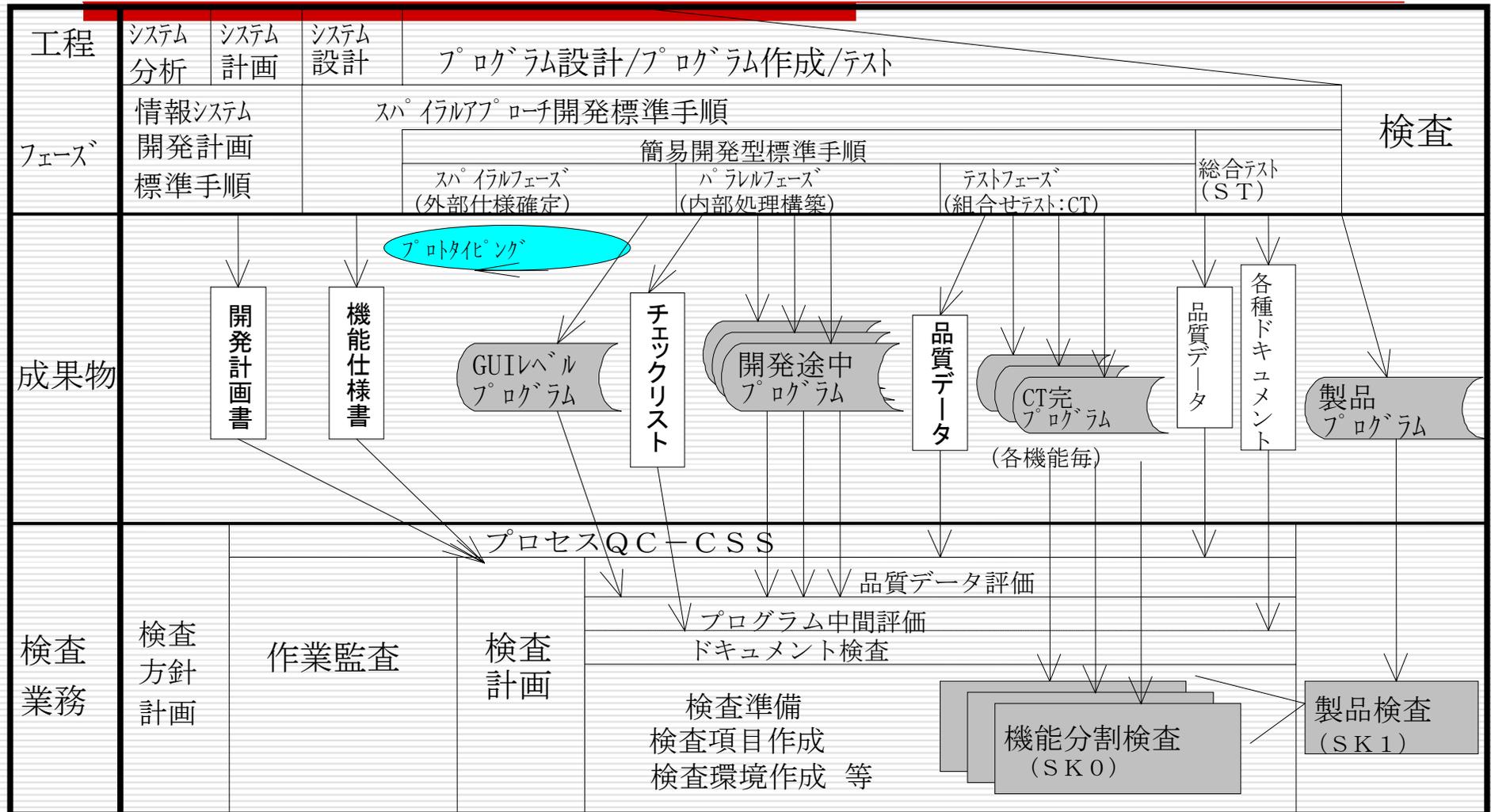
- 多様な開発環境の出現(提案)→簡単、短時間がキーワード
 - 銀の弾丸と錯覚させられた？
 - 一過性のものが多数出現
 - 現在も生き残っているのは???

オープンシステムの開発モデルの例



オープンシステムのRAD型開発モデル

オープンシステムの標準検査作業(90年甲午以降)



オープンシステム対応標準検査作業

オープンシステムの目標値設定の具体例

バグ抽出／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	1.CCL件数: 1.0件/FP	1.SCL件数: 0.4件/FP
2.バグ抽出件数: 0.34件/FP	2.バグ抽出件数: 0.04件/FP	2.バグ抽出件数: 0.02件/FP
非手続き型言語への対応		

* PCL : 単体テスト項目、CCL : 結合テスト項目、SCL : 総合テスト項目

‘90年代 技術の変化 (出展:日経システム2009, 2)

□ ERP(Enterprise Resource Planning)の時代

■ システム構築技術

WEB, ERPパッケージ

■ 開発言語

Visual Basic、第4世代言語

■ インフラ

サーバ(C/Sシステム)、PC

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組み ***

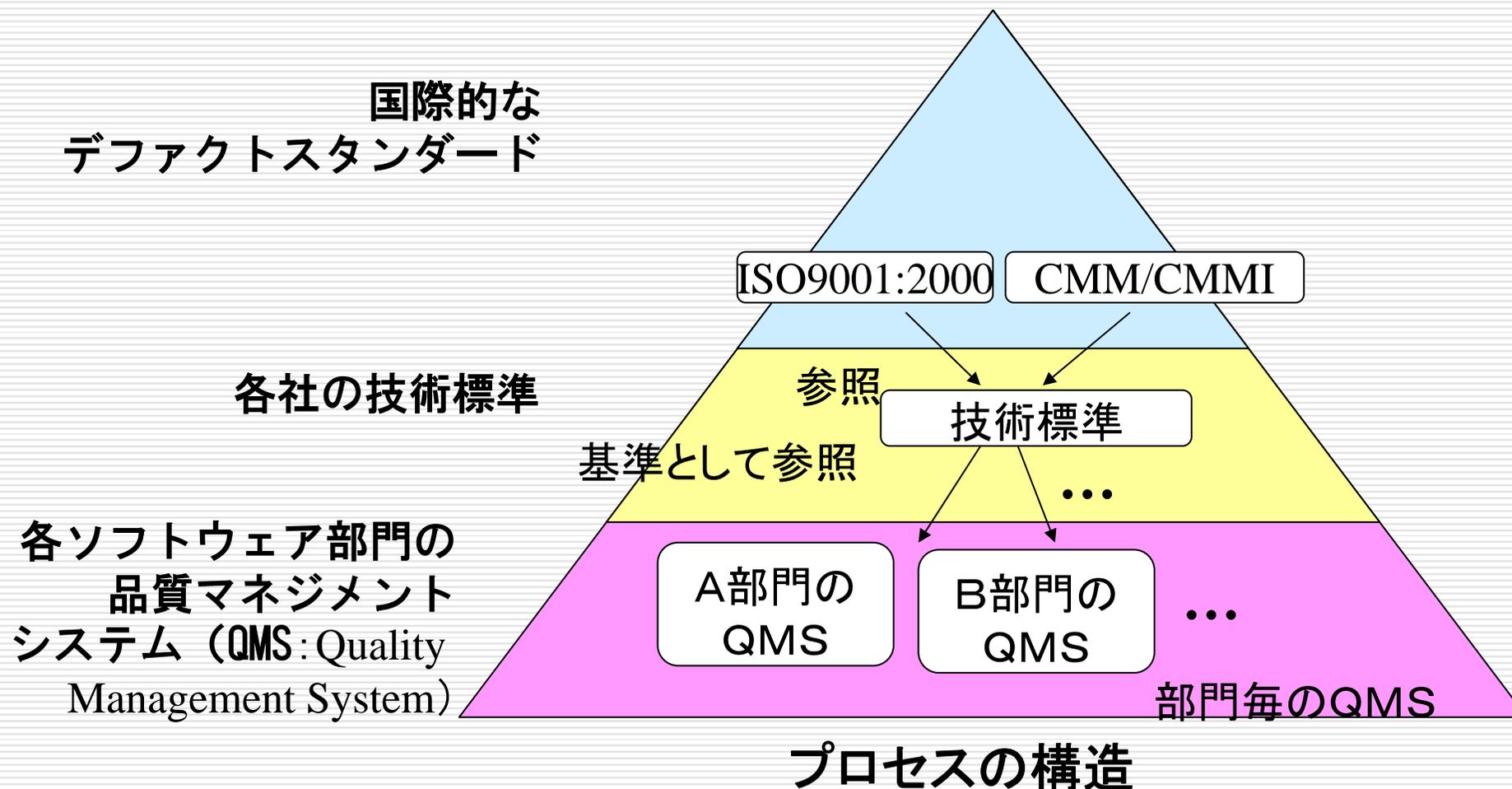
- モダンPMへの本格的な適用
 - 本格的PMSの開発と適用、ノウハウの蓄積と活用

- SPIへの本格的な取組み → 認証取得からの脱却
 - ISO 9001-2000（品質マネジメントシステムへ）

 - CMM Iの発行を契機に、真に効果的適用方法、改善手法の模索

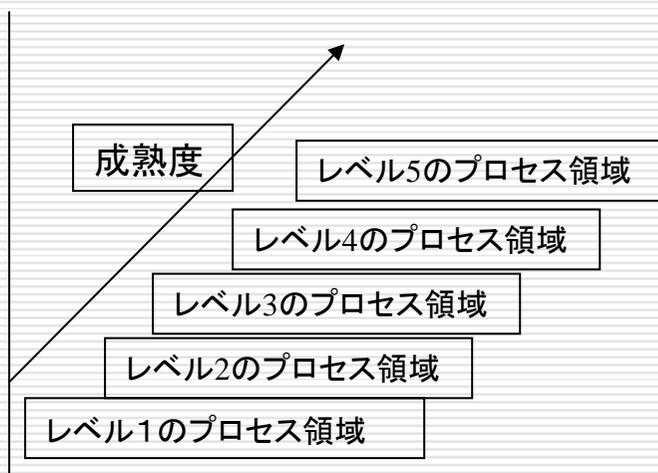
 - PPQAへの本格的な取組み
 - 正しく作ったか、正しく作られたか、の両立性
 - プロセス VS プロダクトからの脱却

品質保証プロセスの構築



出典: 小笠原他、東芝におけるSPI活動事例～現在・過去・未来～、第3回ソフトウェアプロセス改善
ワークショップ(JASPIC主催)、2006.6

CMMIの表現形式



段階表現

レベル5	最適化している
レベル4	定量的に管理された
レベル3	定義された
レベル2	管理された
レベル1	初期



連続表現

レベル5	最適化している
レベル4	定量的に管理された
レベル3	定義された
レベル2	管理された
レベル1	実施された
レベル0	不完全

CMMI : 段階表現モデル

5つの成熟度レベル (ML : Maturity Level) で構成される



ML5 最適化している (継続的改善)

ML4 定量的に管理された (定量的技法)

ML3 定義された (標準化され記述される)

ML2 管理された (計画、実施、測定、制御される)

ML1 初期 (優秀な個人に頼っている・場当たりの)

成熟度レベルとプロセス領域

	エンジニアリング	プロセス管理	プロジェクト管理	支援
ML5		組織改革と展開(OID)		原因分析と解決(CAR)
ML4		組織プロセス実績(OPP)	定量的プロジェクト管理(QPM)	
ML3	要件開発(RD) 技術解(TS) 成果物統合(PI) 検証(VER) 妥当性確認(VAL)	組織プロセス重視(OPF) 組織プロセス定義(OPD) 組織トレーニング(OT)	統合プロジェクト管理(IPM) 統合チーム編成(IT) リスク管理(RSKM)	決定分析と解決(DAR) 統合のための組織環境(OEI)
ML2	要件管理(REQM)		プロジェクト計画策定(PP) プロジェクトの監視と制御(PMC) 供給者合意管理(SAM)	測定と分析(MA) 構成管理(CM) プロセスと成果物の品質保証(PPQA)

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

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SPC, 品質改善の復活
(現場に根ざした改善活動)
- 各種レビューの適用、高度化(SWインスペクション、など)
- メトリックス(バグ、規模、工数、その他)の採取と活用
- V字モデルに基づくフェーズドテストの徹底
- 基本的な管理手法の徹底(バグ管理図の適用など)
- テストにおける品質改善プロセスの採用

レビュー(検証)の充実 ～ソフトウェアインスペクション～

(1) 成果物を第三者が厳密なルールにより検証し、欠陥を除去する。さらにその結果を品質管理、プロセス改善、合否判定へと繋げる。

□ 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

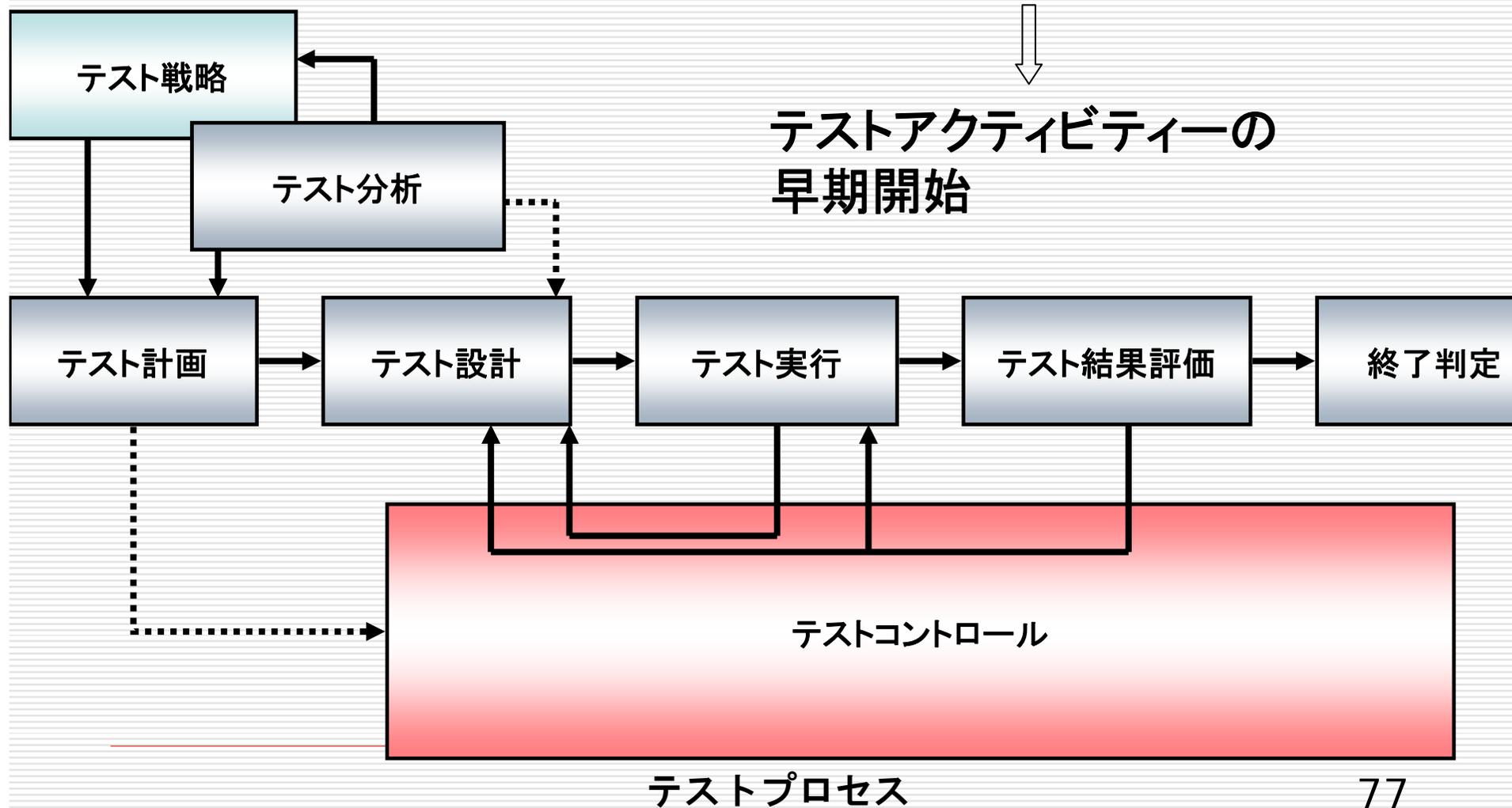
□ 基本的な方法はウォークスルーと類似、但しルールは厳密

□ モデレータと呼ばれる実施責任者が、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、類似エラーの抽出を支援)

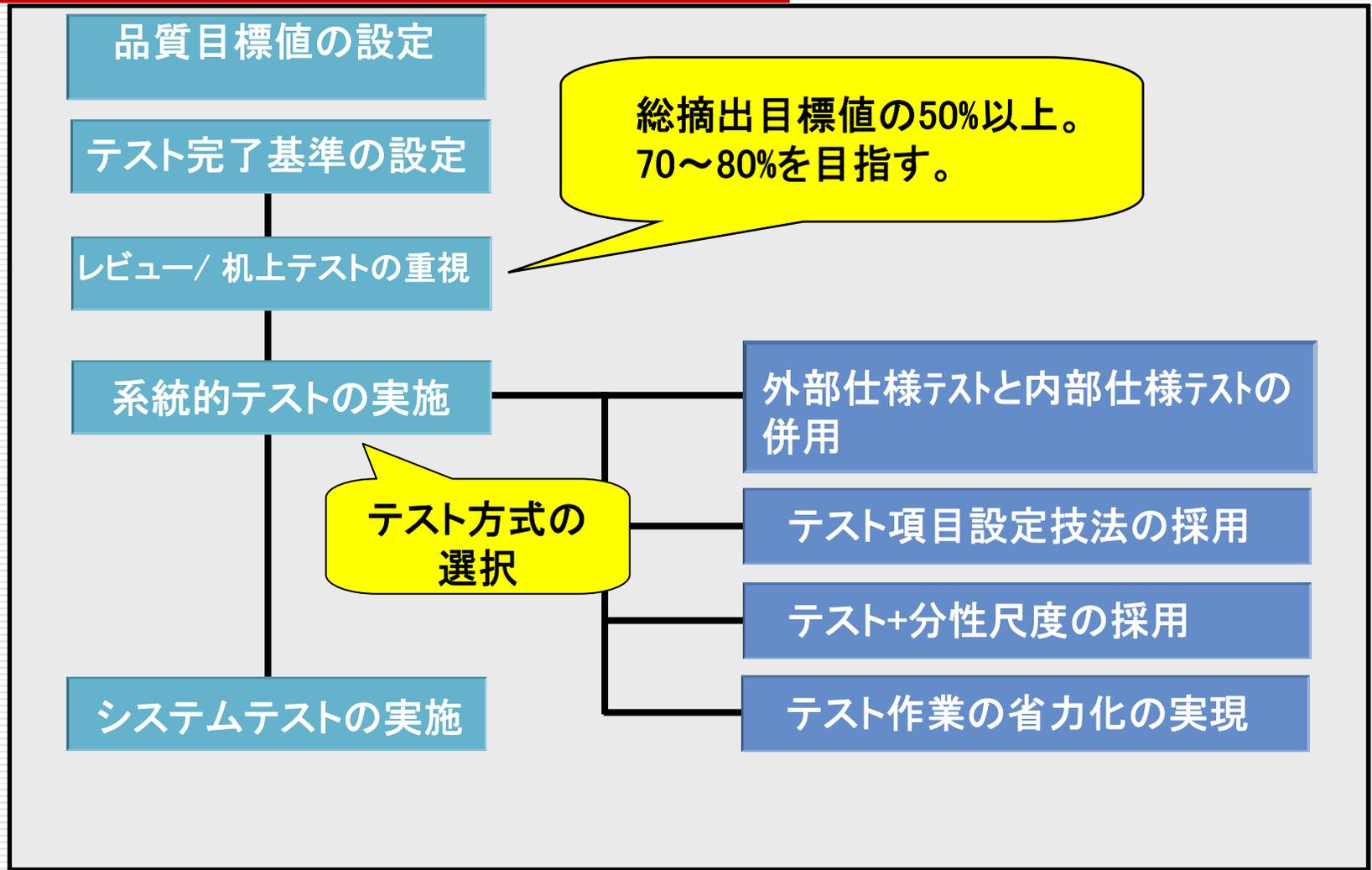
ソフトウェアテスト重要性の再認識

テストプロセスの定義



補足1:テスト戦略策定の例

テスト分析が必要！



補足2: テスト分析とテスト設計の視点

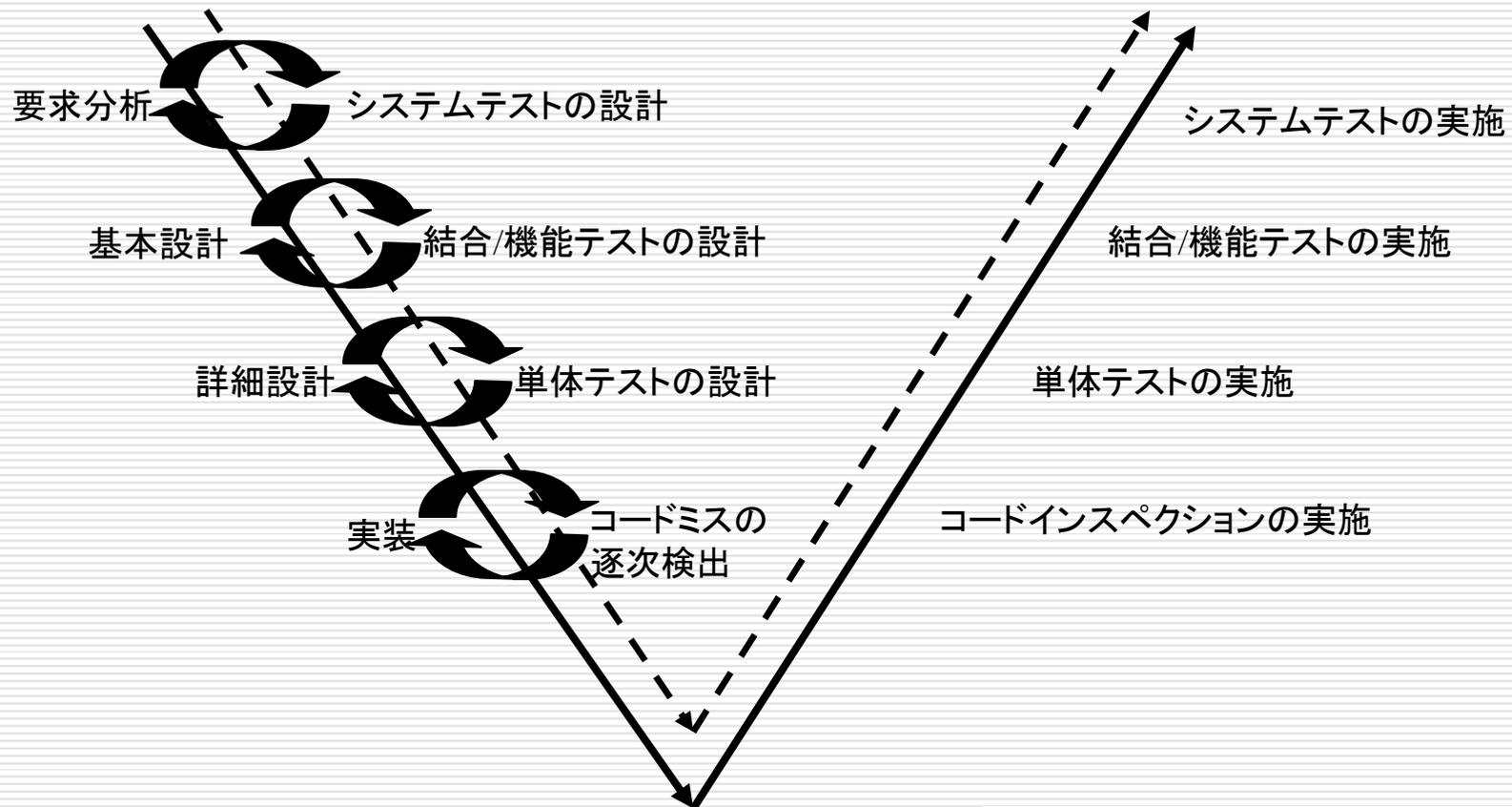
(1) テスト分析の視点

- ① ソフトウェアの重要性 (社会的影響、使われ方 / 局面、複雑さ)
- ② 開発の難易度
- ③ 開発方法論
- ④ 開発期間

(2) テスト設計の始点

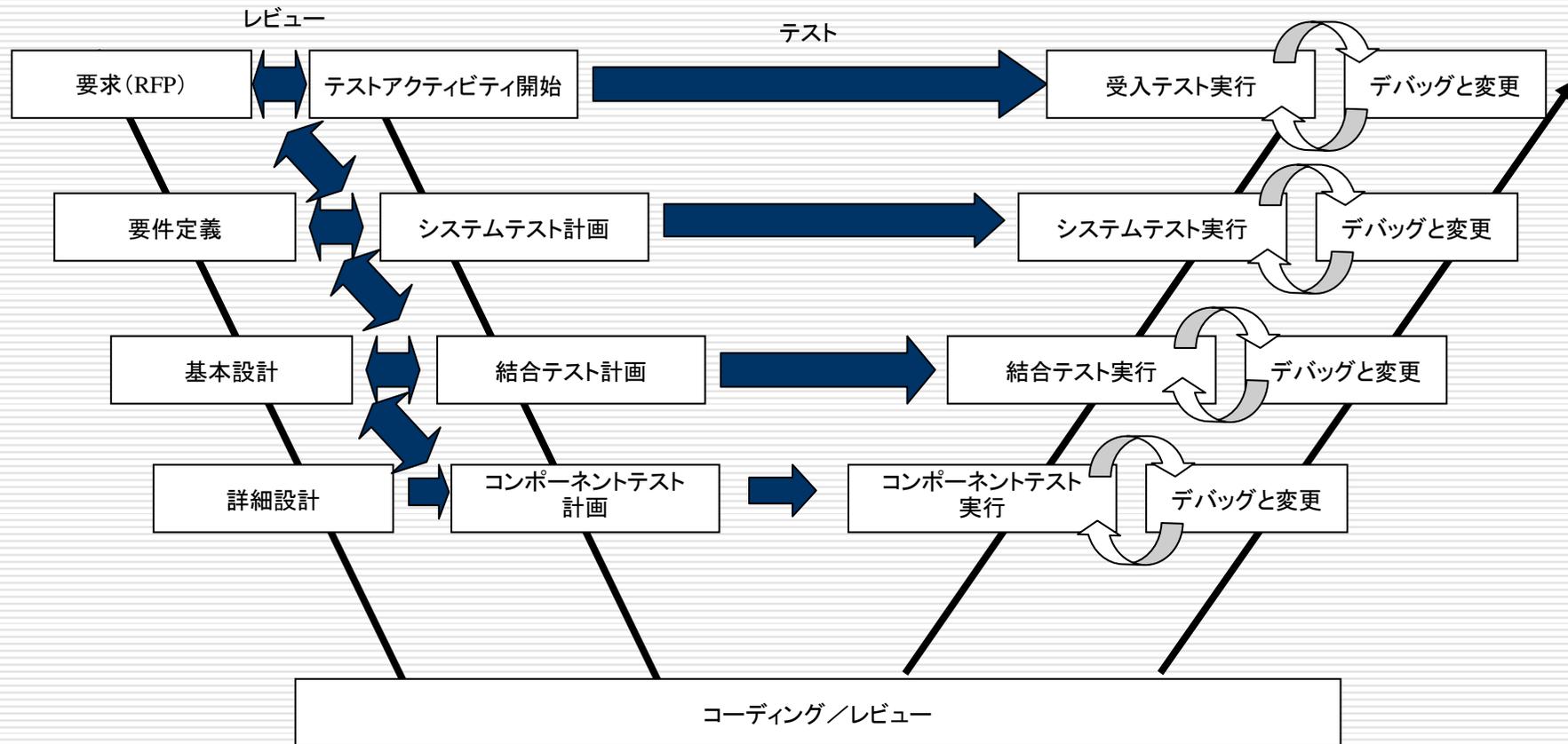
- ① ユーザの視点 (どう使うか)
- ② バグの視点 (バグで困ること、影響)
- ③ 仕様の視点 (何を、何故)
- ④ 設計の視点 (どのように実現したか)

ダブルV字モデル



電通大 西先生 講演資料より

開発のプロセス: W-mode I



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

Wモデル

テストのトピックス

1. ISTQB Foundation Level シラバス、用語集
(技術標準:ISTQB)
2. マインドマップから始めるソフトウェアテスト(書籍:技術評論社)
3. ソフトウェアテスト HAYST法(書籍:日科技連)
4. オールペア法とPICT(技法:ボランティア(WEB))
5. TestLink(ツール:TEF)
6. TEF (Testing Engineer's Forum)(コミュニティ)
7. 現場で使えるソフトウェアテスト(Java編)(書籍:翔泳社)
8. ソフトウェアテスト入門(書籍:技術評論者)
9. ソフトウェアテストの技法 第2版(書籍:近代科学社)
10. ソフトウェアテストの基礎(ISTQB-シラバス準拠)(書籍:BNN新社)

2000年代 技術の変化 (出展:日経システム2009, 2)

□ SOA(Service Oriented Architecture)の時代

■ システム構築技術

フレームワーク、オープンソース

■ 開発言語

JAVA、C#

■ インフラ

サーバ(Webシステム)、PC,モバイル端末

SW品質保証活動の再生に向けて①

- 更なる上流プロセスの強化
 - 創造的仕事の再生
 - 品質作り込みの重視（手戻りのない少ない開発）
- 開発管理の大政奉還（主体を支援部門から開発部門へ）
 - 過剰管理からの脱却
 - やる気になるマネジメントの方法の実現
- ソフトウェア人材の育成（ハイレベルSW工学の習得、専門家集団育成）⇒次のスライド参照
- 「人月の神話（ブルックス）」からの脱却
- 開発技術/管理技術の伝承（積み木崩しからの脱却）
→例: SQuBOK（品質保証技術） ⇒次のスライド参照

人材の育成①

人材の育成

これからの企業における人材育成は平均値を上げる底上げ教育から精鋭を養成する教育と自己啓発を支援する方向に向かうべきであろう。

(1)ソフトウェア工学の基礎理論

開発エンジニアは「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、ソフトウェア開発者の【ディシプリン】を再定義または再確認する必要が有る。

*****一つ例としてPSP(PSP入門)*****

- 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

人材の育成②

(2) 周辺/関連技術の習得

社会学や心理学などの情報システムの開発に必要な分野の知識及びドメイン(業種)知識の習得。

(3) 温故知新的な視点の育成

大きく物事が変化する時代には、過去の流れを見据えて現在から将来を展望できる思考の養成が必要。

(4) 他分野成果の活用能力育成

特にソフトウェア品質管理、品質保証において他分野で成功した技法の吸収と積極的な活用能力の育成。

(5) 情報収集能力の育成

短期開発ニーズが高まりスピードが要求される。これに打ち勝つためには情報収集納能力の向上が必須である。

管理技術の伝承（積み木崩しからの脱却）

ソフトウェア品質の測定とメトリクス

先達の過去の経験とノウハウは「SQuBOK」に纏められている。

こいう物をベースに次の展開を計ることが重要。

□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）」

(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です。

(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つながってはいきません。

(3)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、それらを継続的に改善することです。

SW品質保証活動の再生に向けて②

- 有用な開発技術へのチャレンジ(大企業病からの脱却)
- 革新的管理技術の開発とチャレンジ(大企業病からの脱却)
- テスト技術の更なる進化
- モデル検証の実用化
- 開発経験、失敗経験の共有 → エンピリカル工学？

有用な開発技術へのチャレンジ①

- 段階的(インクリメンタル)開発⇒All in one の回避
開発対象を多数の小さな機能に分割し、1つの反復で1機能を開発する。この反復のサイクルを継続して行い、1つずつ機能追加開発してゆく開発スタイル
- SPLE(Software Product Line Engineering)⇒再利用の新しい道
 - 作らずして創る方向へ
 - '90のコンポーネントウェアは何処へ行った？
- 保守開発、派生開発の斬新的技法の開発
ソフトウェア保守からソフトウェア進化へ
 - ITIL,SOAは有効な方法論になりえるか

有用な開発技術へのチャレンジ②

- モデル検証を前提とした開発
作る前に実現性と有効性を検証
- 形式手法(形式記述言語)
仕様書の記述に形式記述言語を使用し、仕様のあいまい性を排除する⇒如何すれば定着するか
- Eclipse⇒プラットフォームの統合を目指して
Javaの統合開発環境。オープンソース

革新的開発管理技術へのチャレンジ

残念ながらノーアイデア

～ソフトウェア開発には何らかのマネージメント、
管理が必要～

- 従来のマネージメントするからリードするへ
- モチベーションを尊重したワクワクする管理へ

2010年代 技術の変化 (出展:日経システム2009, 2)

□ 事業環境の変化対応への関心が高まる

■ システム構築技術

システムの組合せによる
サービスの開発や品質への
関心が高まる？

著書紹介「ソフトウェア品質保証入門」

ソフトウェア品質保証入門

高品質を実現する考え方とマネジメントの要点



保田 勝通著

奈良 隆正著

A5判 248頁

定価 2940円 (税込)

ISBN978-4-8171-9263-9

出版社：(株)日科技連出版社

URL <http://www.juse-p.co.jp>

私に申し込んで頂くと著者割り20%で出版社からお送りします。
その際は、ご芳名、お送り先住所、TEL、部数をお知らせ下さい。

【連絡先】

〒240-0113 神奈川県三浦郡葉山町長柄1303-36 奈良隆正

TEL/FAX:046-875-6711

E-mail:nara@consul21.com

またはt-nara@mx4.ttcn.ne.jp

本書は、「組織としてソフトウェアの品質保証をどのように確保すれば良いのか」という課題に“初めて”品質保証を担当する方々にもわかるように、基礎的な知識からいねいに解説しています。今後、ますます激化するグローバルな競争の中で生き残るために「何を」、「どうすべきか」についても解説します。さらに、品質を確保するために役立つ、ツール、手法、指標、メトリクスなどについて例をあげながら紹介します。

新 知 故 温

そして

新 創 顧 温

ご清聴ありがとうございました