

# 「品質会計」構築の歴史にみる ソフトウェア品質保証のありかた

2010年 11月29日

日本電気株式会社

誉田 直美

「品質会計」は、日本電気(株)の登録商標です。

人と地球にやさしい情報社会を  
イノベーションで実現する  
グローバルリーディングカンパニー

**NECグループビジョン2017**

# 書籍のご紹介 「ソフトウェア品質会計」

## ソフトウェア品質会計

～ NECの高品質ソフトウェア開発を  
支える品質保証技術～

誉田 直美著

2010年6月30日発行

日科技連出版社

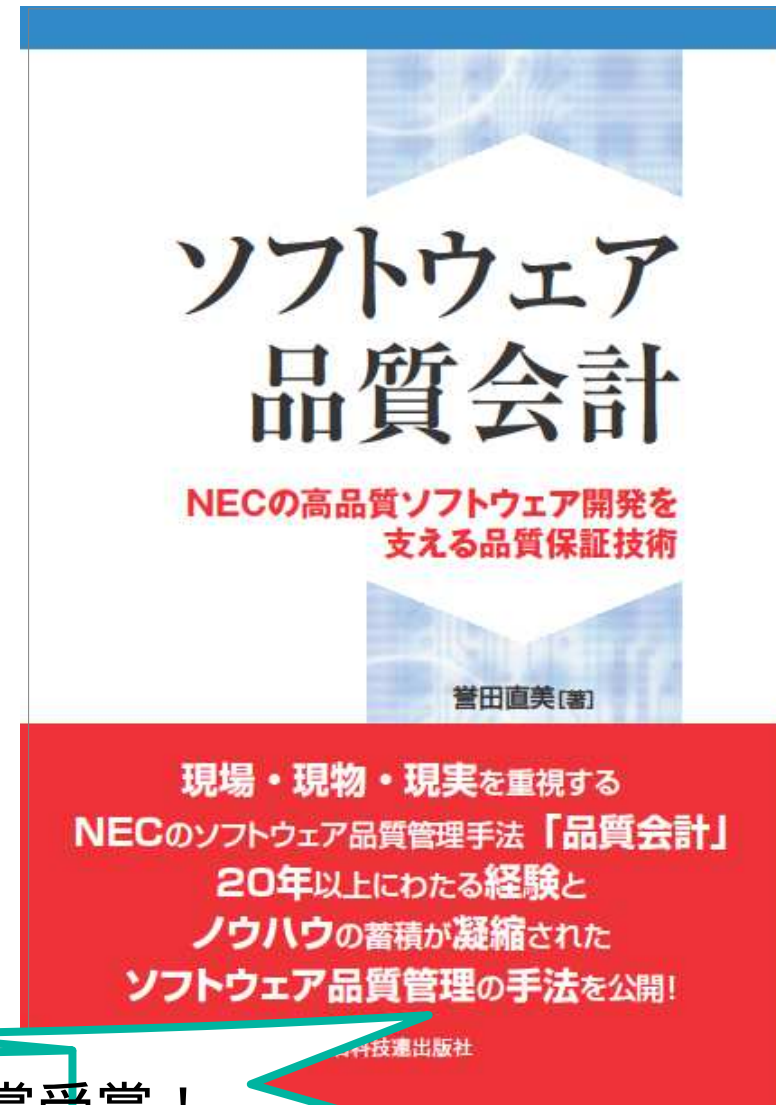
定価 3150円(税込)

ISBN978-4-8171-9348-3

品質会計は、NEC独自のソフトウェア品質管理手法であり、NECグループで広く適用されています

ケーススタディを含め、その使い方を具体的に解説しました

現場・現物・現実を重視している点が、大きな特徴です



2010年度 日経品質管理文献賞受賞！

# 事業領域と主な商品・サービス

<p>ITサービス事業</p>	<p>ITプロダクト事業</p>  <p>WebSAM</p>	<p>パーソナルソリューション事業</p>  <p>docomo STYLE series* N-018</p>
<p>ネットワークシステム事業</p>  <p>UNIVERGE</p>	<p>社会インフラ事業</p> 	
<p>エレクトロニクスデバイス事業</p> 		

1. 品質会計とは
  
2. ソフトウェア品質保証のありかた
  - ・品質保証機能の変遷
  - ・成果の差異はどこから生ずるか
  - ・ソフトウェアの品質保証において重要と思うこと

---

# 1. 品質会計とは

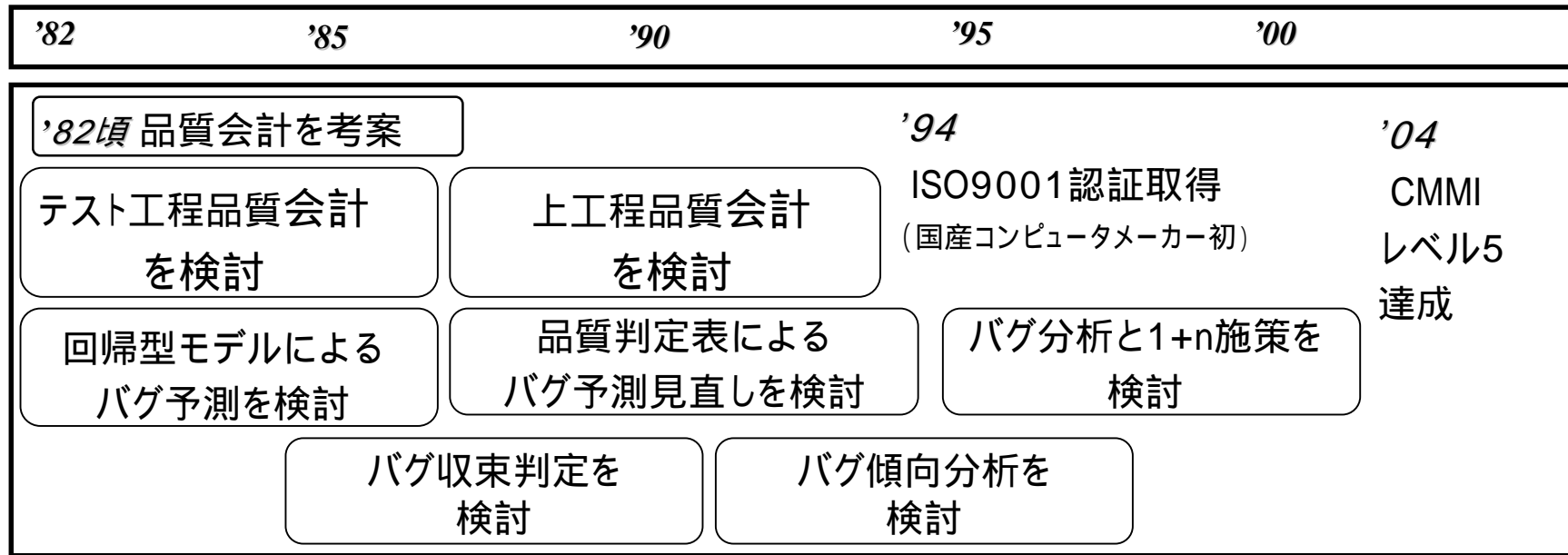
# なぜ、品質会計は生まれたか

- 果て知れぬフィールドバグとの戦い...
- 負のスパイラルのままでは事業継続できない(徹夜続きで体が持たない)との思いから、現場が考案し発展



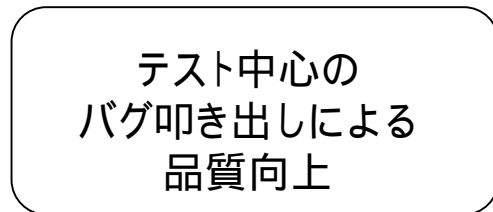
ダンプとトラブルの山に埋もれ.....

# 品質会計発展の歴史

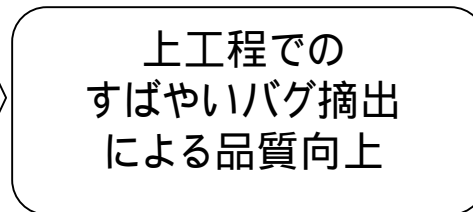


[品質改善の視点]

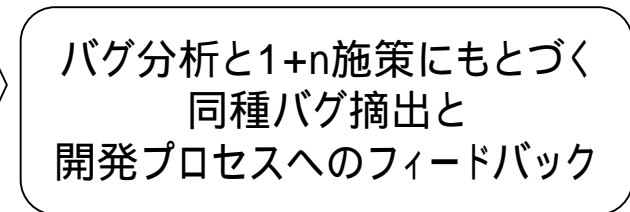
< 第1期 >



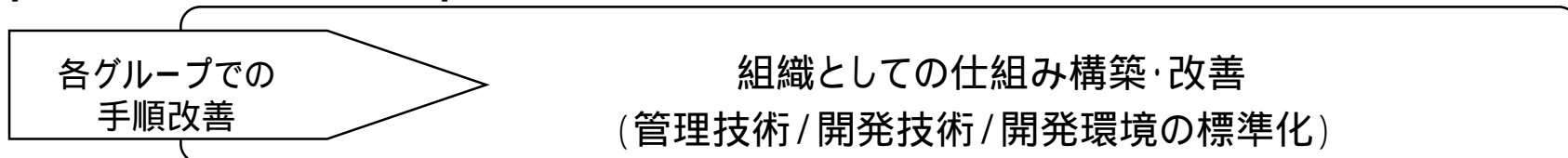
< 第2期 >



< 第3期 >



[開発プロセス構築の視点]





# 品質会計: Quality Accounting System とは

---

■ 「品質」が作り込まれたことを、確かな根拠をもって 説明するソフトウェア品質管理手法

- “account”とは、もともと「理由・根拠を説明する」という意味を持つ

- 品質会計を特徴づける技法

- 上工程品質会計

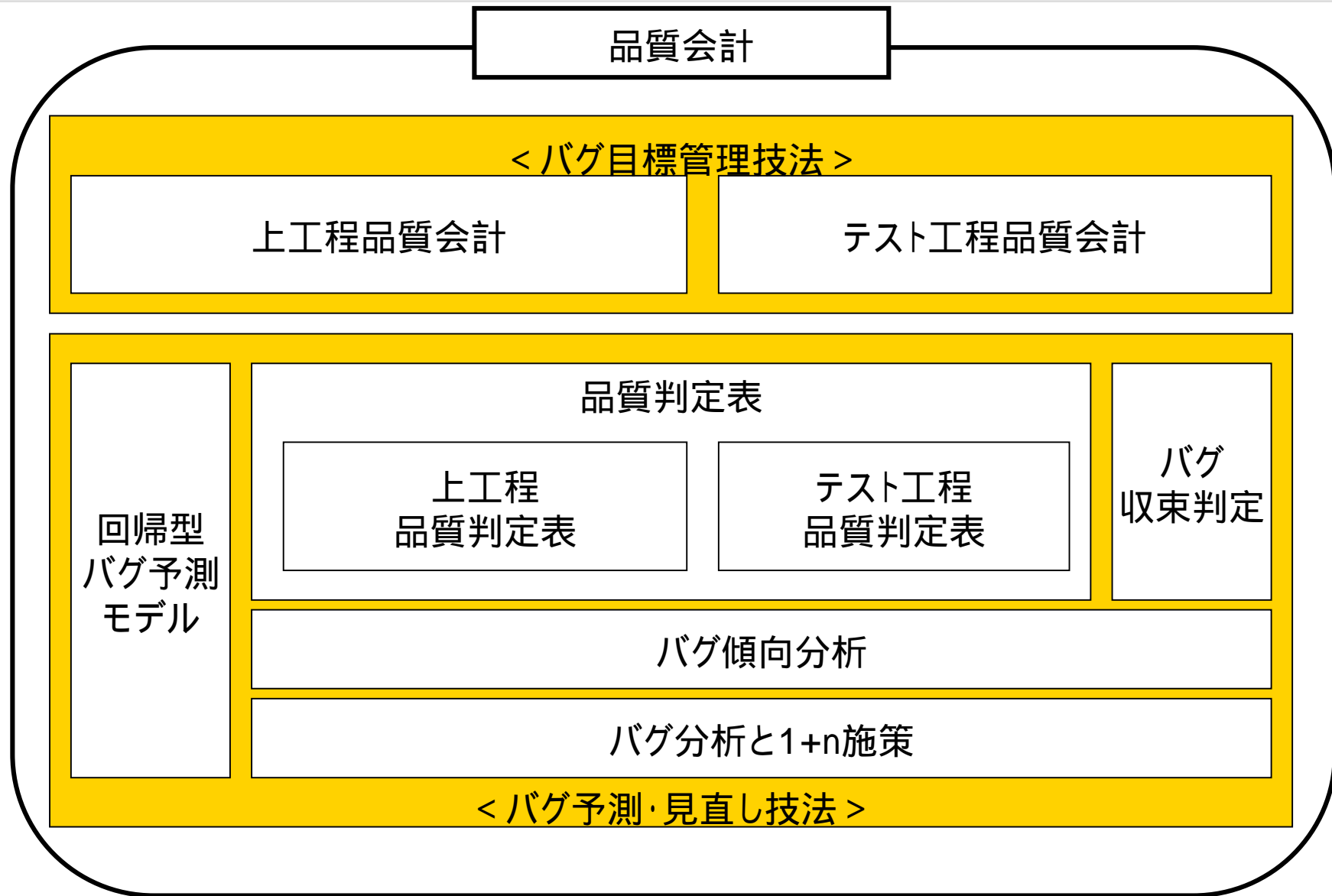
- テスト終盤の残存課題の分析

- バグ傾向分析、バグ分析と1+n施策、バグ収束判定の3つの組み合わせ

■ 1982年頃、NECの開発現場で考案された、NEC独自の手法

■ 「作り込んだバグを負債と見なし、バグ摘出によりこの負債を返済し、負債がゼロになった時点で出荷する」ことを基本とする

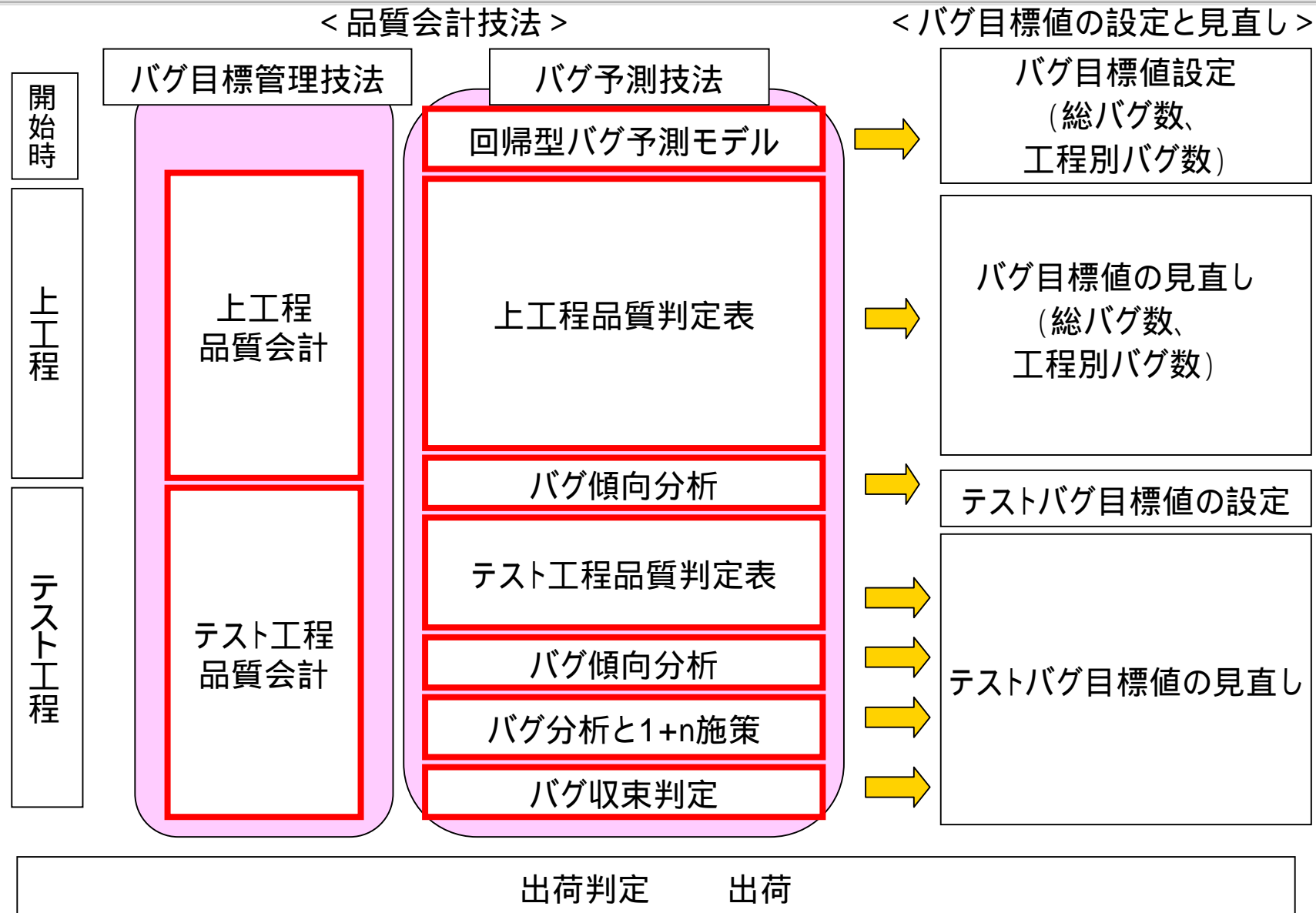
# 品質会計の技術体系



# 品質会計を構成する技法とその特徴

カテゴリ	技法	使用法と特徴
バグ目標管理技法	上工程品質会計	上工程(設計～製造)用のバグ目標管理技法。バグの抽出工程と作りこみ工程の両面からバグを目標管理する。
	テスト工程品質会計	テスト工程用のバグ目標管理技法。テスト開始時に残存する、プログラム全体の総バグ数を目標管理する。
バグ予測・見直し技法	回帰型バグ予測モデル	開発開始時に、今回の開発で作り込むであろう総バグ数を予測するためのバグ予測技法。
	品質判定表	開発途中に発生する変化を考慮して、バグ目標値を見直すバグ目標値見直し技法。上工程品質判定表とテスト工程品質判定表がある。
	バグ傾向分析	抽出したバグをさまざまな観点から整理することにより、バグの抽出傾向に偏りが無いかを分析する技法。
	バグ分析と1+n施策	バグ1件ごとに真因を分析することにより、開発上の細かい抜け・漏れを発見し、その抜け・漏れに対して、集中的なレビューやテストにより残存するバグを抽出する技法。 バグ分析と1+n施策はセットで用いる。
	バグ収束判定	テスト度合いに対する累積抽出バグ数の推移により、バグ収束を判定する技法。

# 開発プロセスと品質会計技法の適用の関係



# 品質会計の原則

## < 品質会計の原則 >

■ バグは作り込まない。作りこんだバグはすばやく摘出する。

## < 上工程品質会計の原則 >

■ 作り込んだバグは次工程までに摘出する

- 作り込み工程で80%摘出
- 次工程で残り20%摘出

## < テスト工程品質会計の原則 >

■ 作り込んだバグは、すべて摘出してから出荷する

## < 目標 >

■ 上工程バグ摘出率 80%

# 上工程品質会計とは

---

## 特徴

- 上工程(設計～コーディング)用のバグ目標管理技法
- バグの摘出工程と作り込み工程の両面からバグを目標管理
- 品質会計を代表する技法

## 目標管理の視点

- バグ摘出工程別の管理
  - ・ 開発の経緯につれて、どのようにバグを摘出しているか
- バグ作り込み工程別の管理
  - ・ どの開発工程の作業内容に問題があるか

上工程品質会計は、品質向上の大きな原動力

# テスト終盤の残存課題の分析の考え方

## バグ傾向分析

大きな抜け漏れは  
ないか

3点を満足  
= 残存課題なし

## バグ収束判定

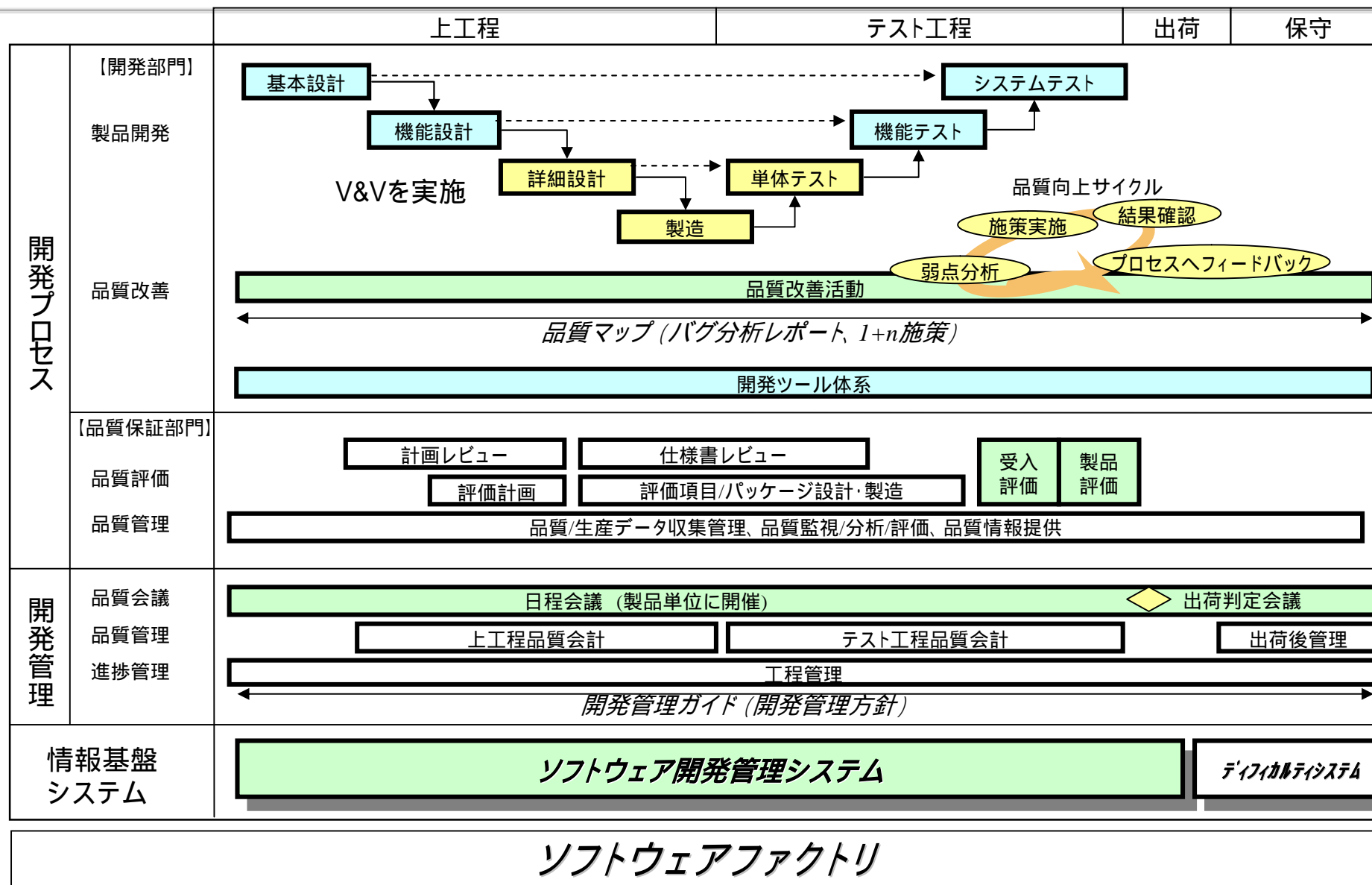
実際にバグは  
もう出ないか

細かい抜け漏れは  
ないか

## バグ分析と1+n施策

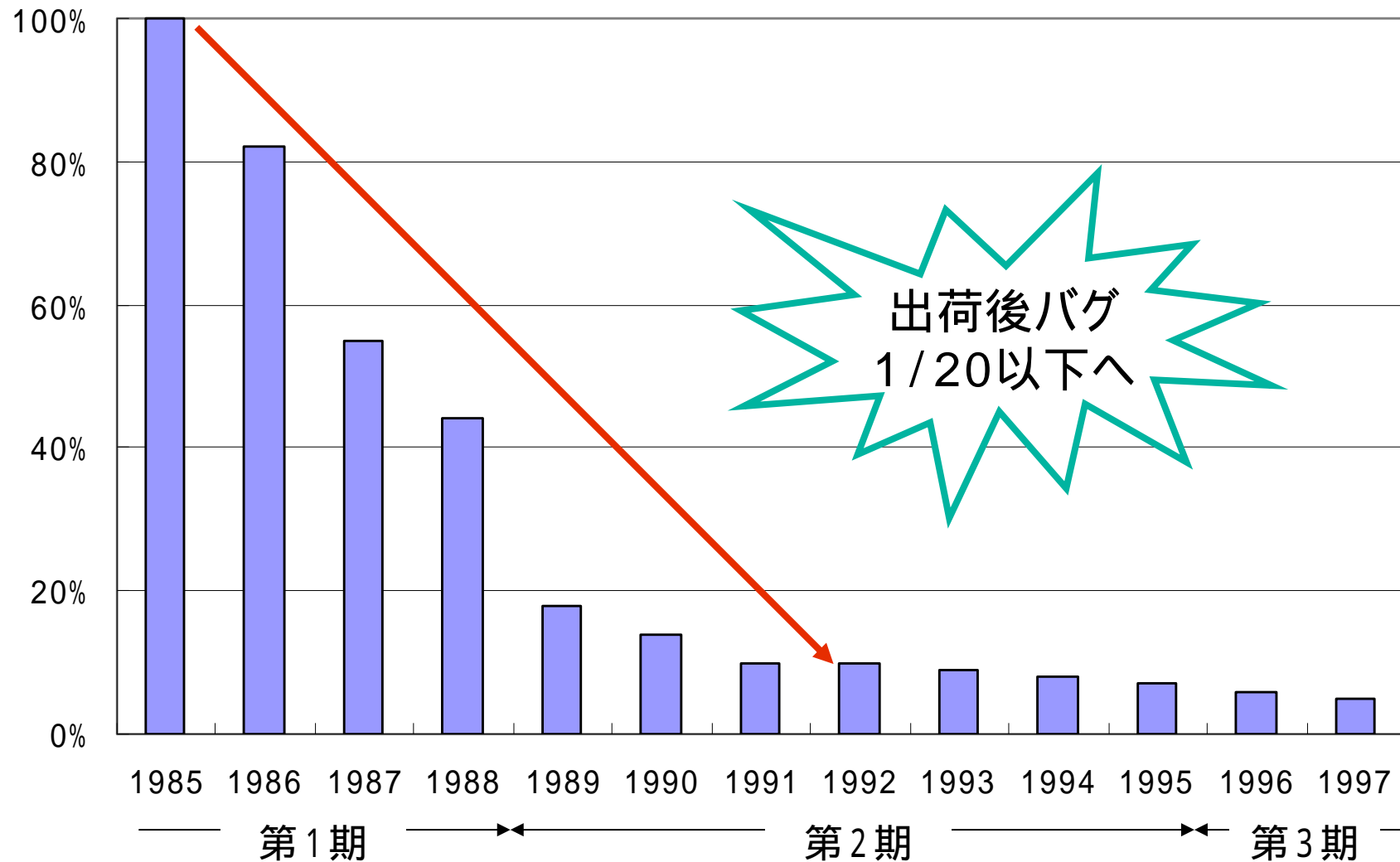
テスト終盤までプロセス通り  
実施されていることが前提

# 品質保証体系：全体的なしくみで品質確保





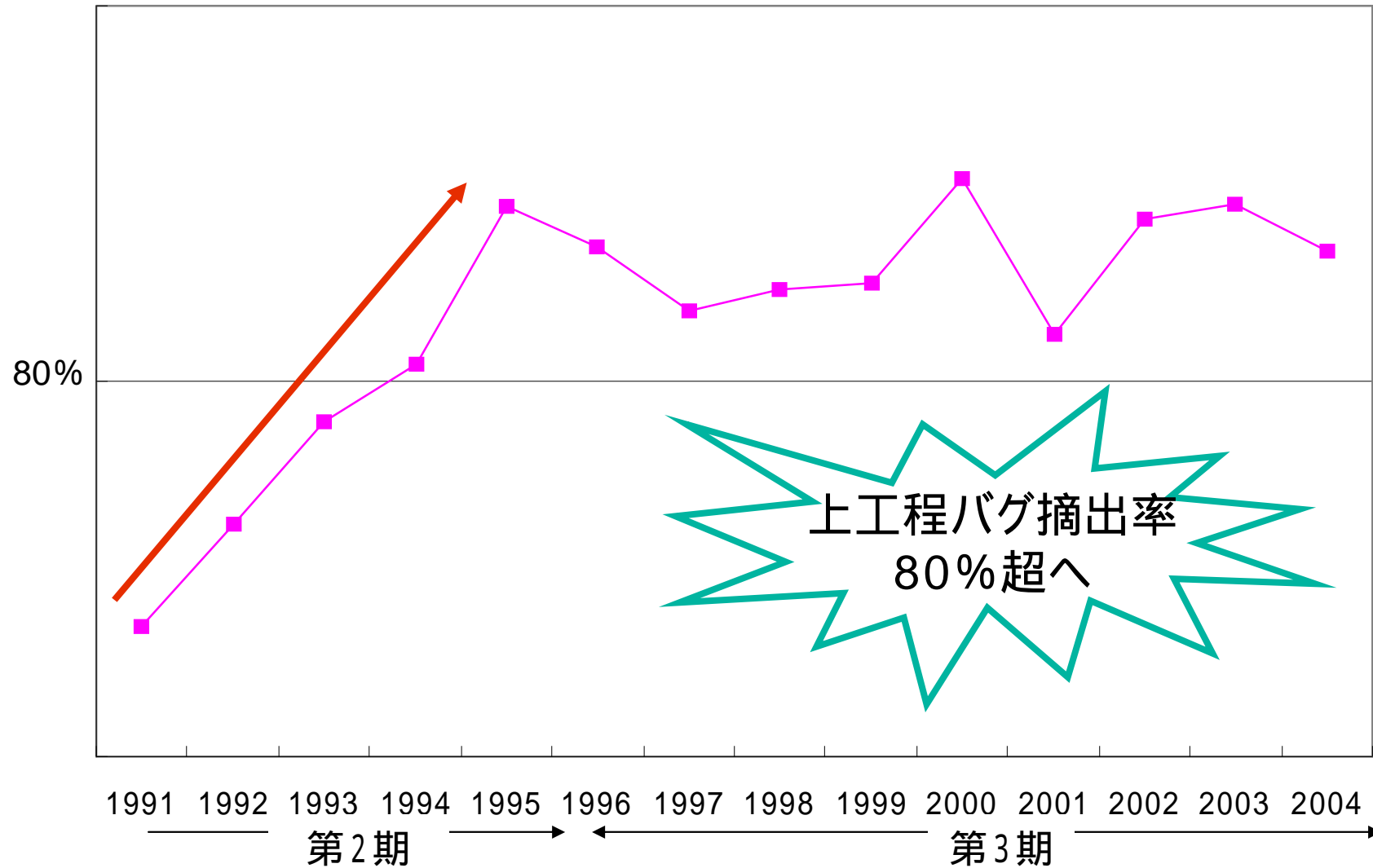
# 品質会計の効果：出荷後バグの削減推移



1985年の出荷後バグ件数を100%とした相対比を表示

# 品質会計の効果：上工程バグ摘出率の推移

上工程バグ摘出率 (%)



---

## 2. ソフトウェア品質保証のありかた

---

# 品質保証機能の変遷

# 品質保証の諸機能

品質保証の機能	内容
③ プロセス評価	プロジェクトのプロセス実施状況进行评估
① プロダクト評価	プロジェクトから出力される成果物を評価
コンサルテーション	プロジェクト状況を客観的に分析し、プロジェクトへ助言
④ プロセス改善	組織のソフトウェアプロセスを改善
② 出荷判定	出荷判定をする(出荷停止権限を保有)
出荷後のトラブル対応	出荷後の客先トラブル対応

③ : 着手した順番

■ : 担当する機能

# 品質会計発展の歴史

再掲

'82	'85	'90	'95	'00
'82頃 品質会計を考案			'94	'04
テスト工程品質会計 を検討		上工程品質会計 を検討	ISO9001認証取得 (国産コンピュータメーカー初)	CMMI レベル5 達成
回帰型モデルによる バグ予測を検討		品質判定表による バグ予測見直しを検討	バグ分析と1+n施策を 検討	
	バグ収 束	QA評価はもう 不要？	分析を 検討	QA評価があるから 納得性が増す

[品質改善の視点]  
< 第1期 >

テスト中心の  
バグ叩き出しによる  
品質向上

< 第2期 >

上工程での  
すばやいバグ抽出  
による品質向上

< 第3期 >

バグ分析と1+n施策にもとづく  
同種バグ抽出と  
開発プロセスへのフィードバック

QA評価と出荷判定

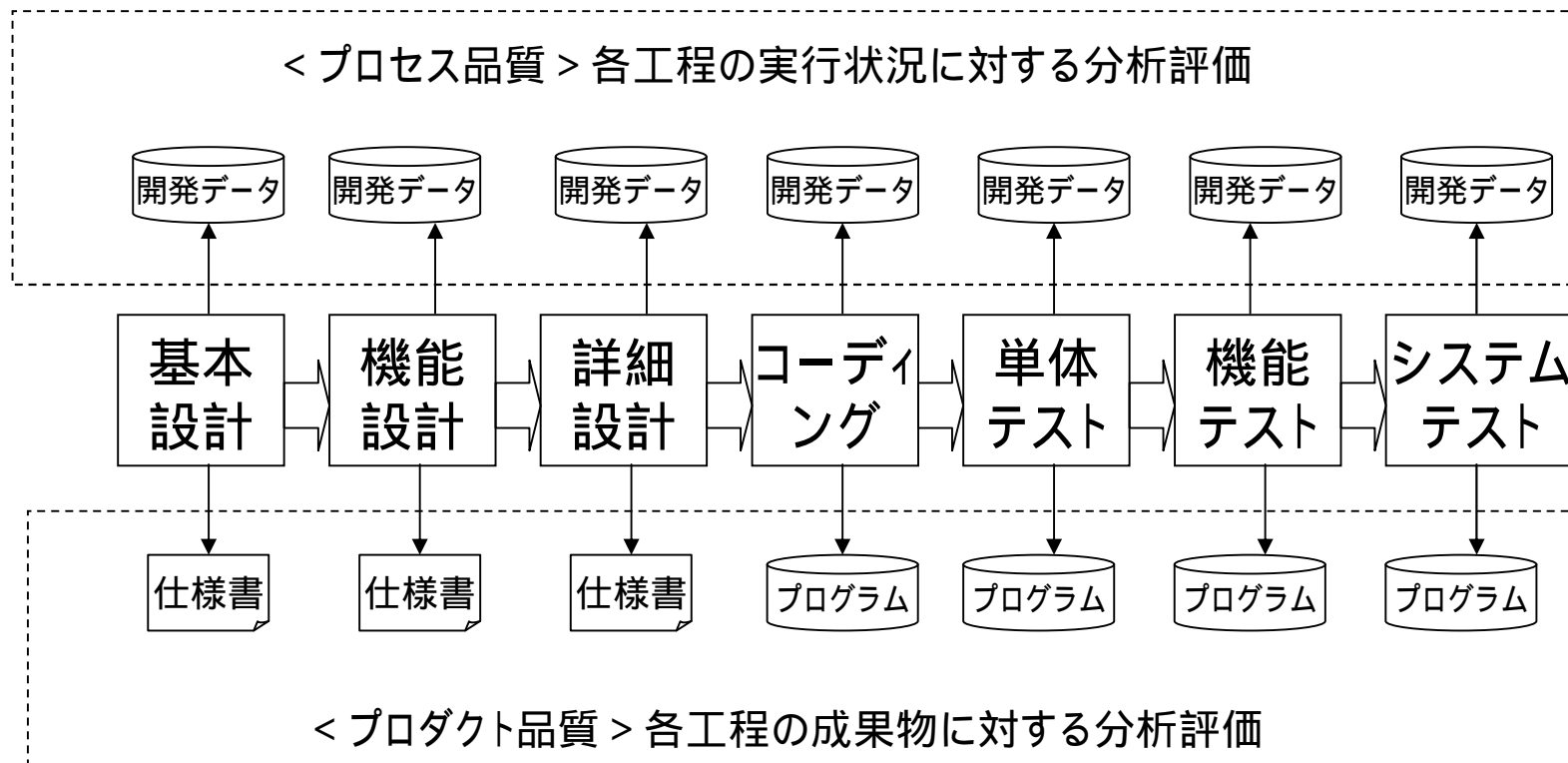
プロセス評価

プロセス改善

各グループでの  
手順改善

組織としての仕組み構築・改善  
(管理技術 / 開発技術 / 開発環境の標準化)

# プロセス評価とプロダクト評価



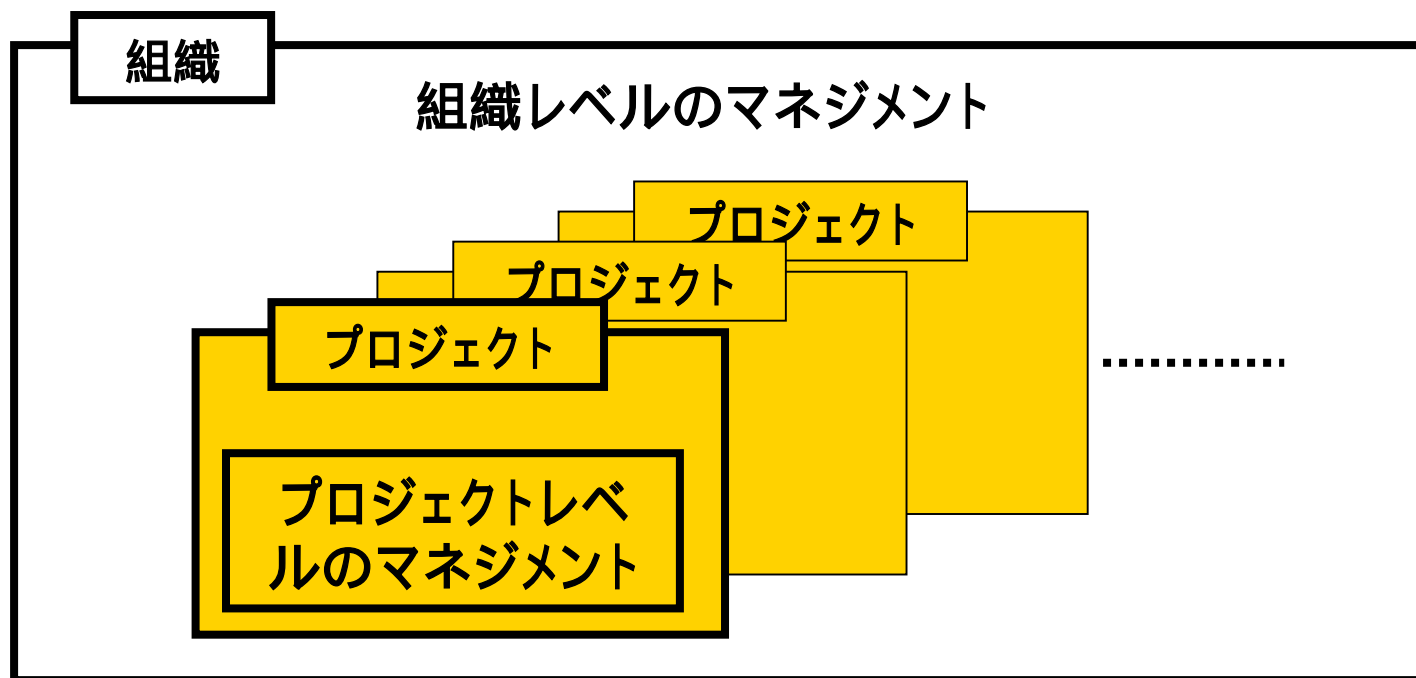
# 組織とプロジェクトの視点の違い

## 組織レベルのマネジメント

- 組織で実施するプロジェクトのQCD成功確率向上を狙う  
繰り返し実行可能な、よいソフトウェアプロセスの構築と運用が鍵

## プロジェクトレベルのマネジメント

- 当該プロジェクトのQCDの成功を狙う  
組織のソフトウェアプロセスをいかにうまく現場に適用するかが鍵
- プロジェクトとは、「独自の成果物またはサービスを創出するための有期活動」 < PMBOK >

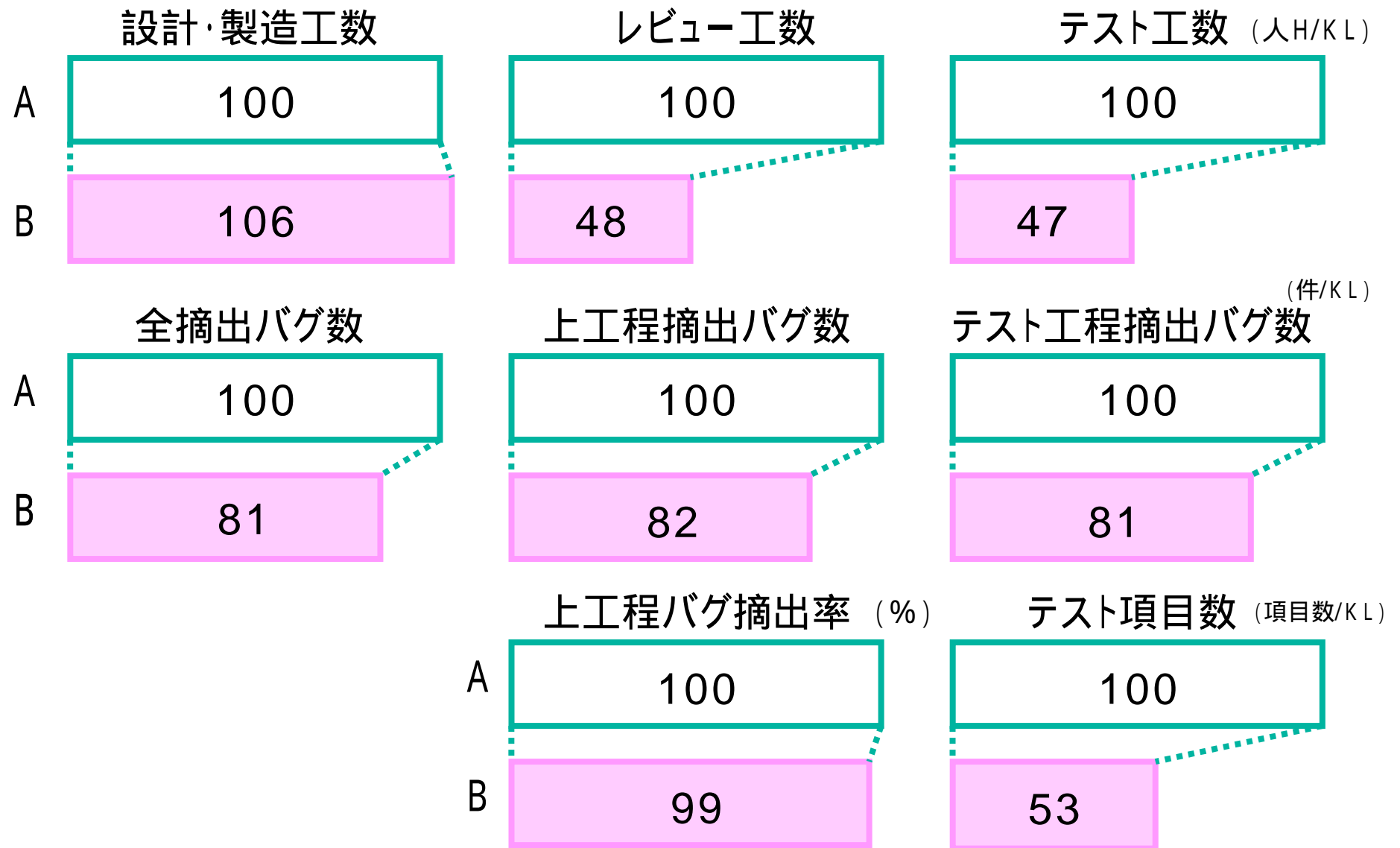




---

成果の差異は  
どこから生ずるか

# どちらの組織のほうが品質が良いか？



- 
- 同じ技法・プロセスを適用しても、同じ効果はでない
  - データだけでは判断を誤る
  - 現場へいかなければ、本当に良いかどうか判断できない

---

ソフトウェア品質保証において  
重要と思うこと

# ソフトウェア品質保証において重要と思うこと

---

成果が出るまで改善し続ける

ぶれない軸足

品質を作りこんだことを説明できる

# 成果がでるまで改善し続ける

品質会計が、開始時のバグ予測 バグ目標値見直しへ軸足を変えるまで

## 回帰型バグ予測モデル

- 開発開始時の予測精度向上に心血を注ぐ わかりやすさ重視

$$B = C \cdot \quad \cdot S^a$$

B : 総バグ数  
C : 標準バグ数  
 : バグへの影響要因  
S : 開発予測規模  
a : 指数係数



$$B = C \cdot \quad \cdot S$$

B : 総バグ数  
C : 標準バグ数  
 : バグへの影響要因  
(主要因はスキル、難易度)  
S : 開発予測規模

「開発途中で要件が変わったり、開発者が交代したり、いろいろなことがあるんだよ。それを考慮したバグ予測にしてもらわなければ困るんだ！」

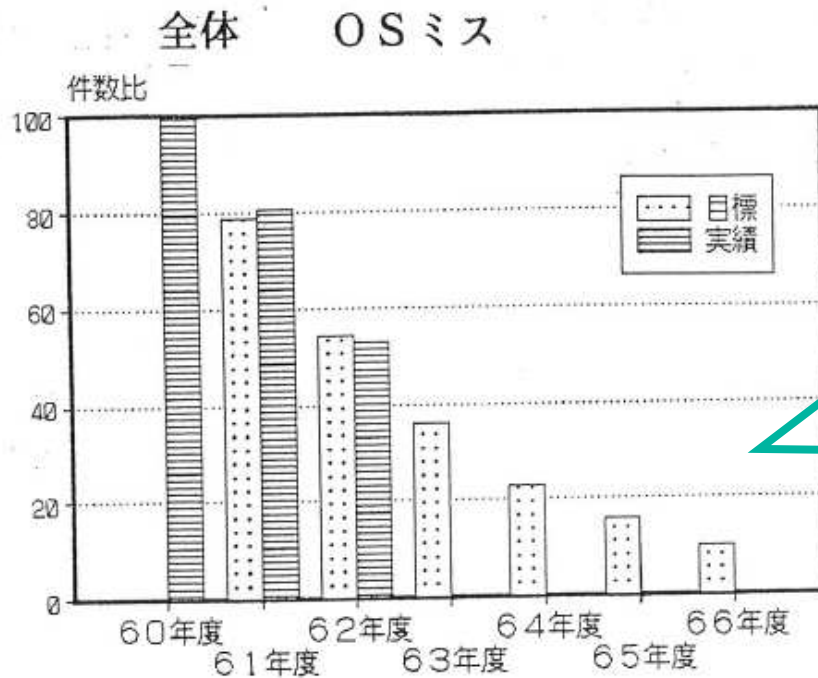
## 開発途中のバグ目標値見直し技術の発展へ

- 開発開始時に、すべてを予測することは不可能
- 開発途中のさまざまな変化を取り込んで、きめ細かく見直すほうが正しい

# ぶれない軸足

さまざまなタイプの人々が、トップになる

- 「1に品質、2に品質、3,4がなくて5に品質」
  - 「コスト削減」「開発は安い海外を使うこと！」
  - 「もうSWQCはいいんじゃないの」
  - 「ウォーターフォールモデルはもう古い、アジャイルだ！」
- こんなとき、品質保証部長はどうする？！



品質会計は、  
品質10倍作戦から  
始まった

# 品質を作りこんだことを説明できる

「品質会計しています」とはどういうことか

×

- 品質会計の帳票にデータを書いて出している
- バグ目標値 = バグ実績値である

(データを出すのは、スタートライン)

- 開発途中の問題と対応を説明できる
- なぜそのバグ目標値でよいのかを説明できる
- 特にテスト最終段階での品質と対応状況を説明できる  
「品質」を作り込んだことを、確かな根拠をもって  
具体的に説明できる



## - 銀の弾丸はない, しかし道はある -

---

■ 技術にせよ管理手法にせよ、単独でソフトウェアの生産性や品質を飛躍的に改善するものはない

■ 王道はない. しかし, 道はある.

- F.ブルックス「人月の神話」より

「品質会計」が単独で効果をだせる範囲は限られる。  
不足している部分を他の技法で補完することが重要。  
(ソフトウェア領域では、すべての技法に同じことが言える)

Empowered by Innovation

**NEC**