

# ソフトウェアメトリクスを活用した品質管理

2010年1月29日

日本電気株式会社  
ソフトウェア生産革新部  
佐伯 剛幸

# 目次

---

はじめに

客観的なソフトウェアメトリクスの必要性

品質管理における後戻り防止の課題

メトリクス適用による品質管理の改善

事例1「予測精度向上による後戻り防止」

事例2「品質の作り込みによる後戻り防止」

コードクローン検出ツールCCFinderXとは

組織の品質管理のための測定ツールの整備

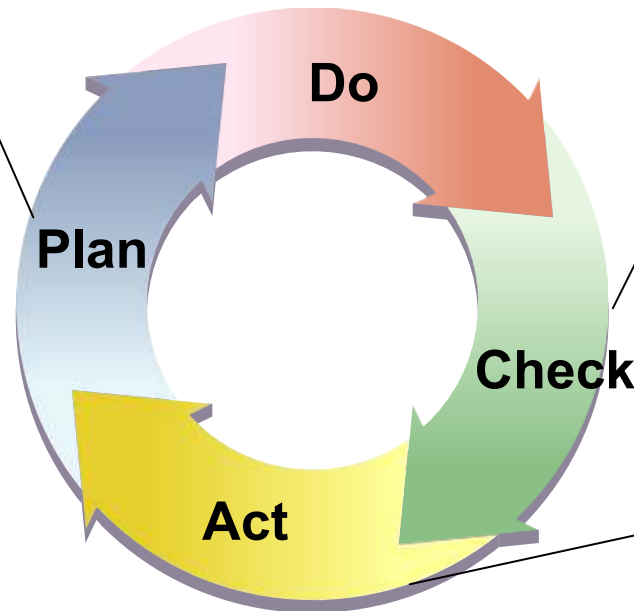
まとめ

# はじめに

本発表では、品質管理のPDCAサイクルにおいて、ソフトウェアメトリクスの活用で開発の後戻りを防止する取り組み事例と運用上の工夫を紹介

## 見積もり・計画

- プロジェクトの工数、日程、品質を高い精度で見積る
- 進行中のトラブルプロジェクトを再計画する
- 組織内の全プロジェクトに対する、リソース割り当ての全体計画を作成する



ソフトウェアメトリクス活用

## 品質・生産性の評価

- 進行中のプロジェクトをコントロールする
- 開発プロセスの改善度合いをモニタする

## 目標値の修正

- 組織の利用するメトリクス標準値を修正する

「初めて学ぶソフトウェアメトリクス～プロジェクト見積もりのためのデータの導き方」日経BP社, 2005 より引用・修正

# 客観的なソフトウェアメトリクスの必要性

■ NECでは、独自の品質会計制度を定め、メトリクスを採取して品質管理

例：開発工数、レビュー時間、ソースコード規模、テスト項目数、バグ数・・・

※ プロセスで得られるメトリクスが多い

■ 管理精度向上のため、より客観的なソフトウェアメトリクスに着目

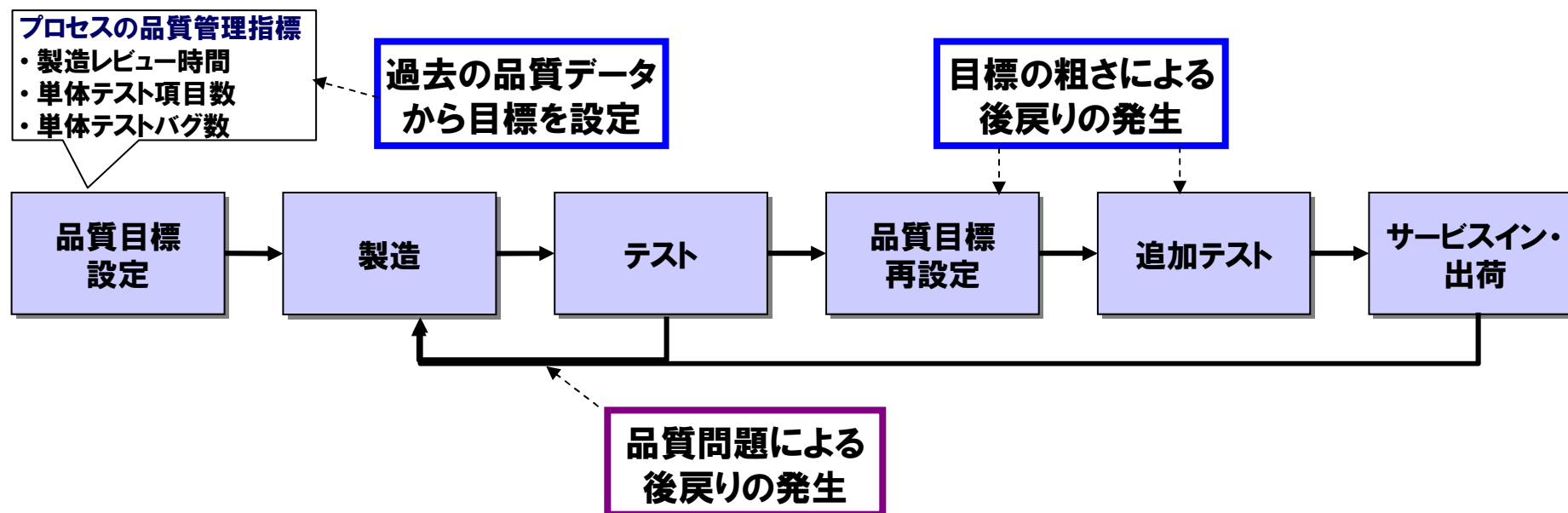
例：複雑度、結合度、クローン数・・・

■ ただし、ソフトウェアメトリクスを扱うツールは、詳細な測定を機械的に行うため結果が大量であり、抽出と分析に工数がかかる

客観的という点に着目してソフトウェアメトリクスを利用、  
大量データからの抽出と分析はツールの利用技術で対処

# 品質管理における後戻り防止の課題

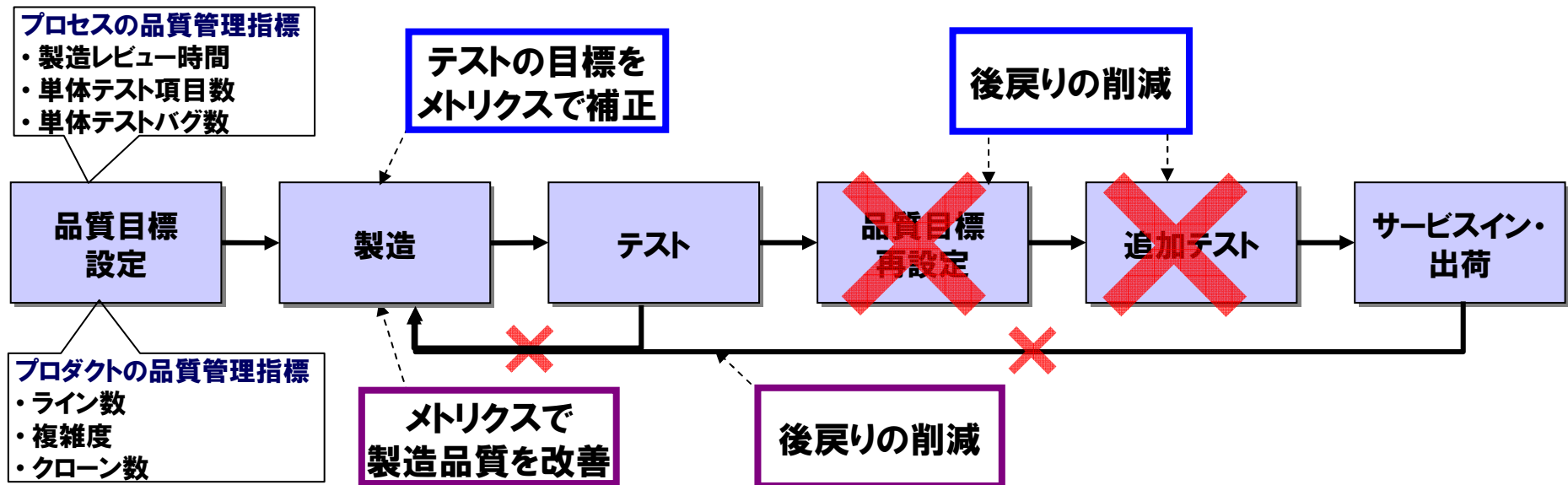
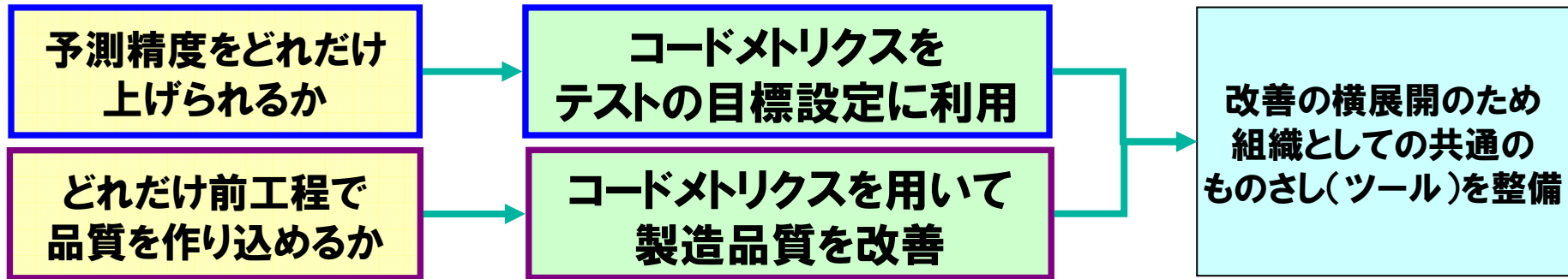
- 予測精度をどれだけ上げられるか
- どれだけ前工程で品質を作り込めるか



# メトリクス適用による品質管理の改善

## 品質管理における課題

## 改善策



※ コードメトリクス: ソースコードを対象とするソフトウェアメトリクスのこと、本発表ではコードメトリクスを主な対象として扱う

# 事例1「予測精度向上による後戻り防止」

---

■ 従来は、開発規模、難易度、開発者スキルと過去の品質データからバグ抽出目標を設定

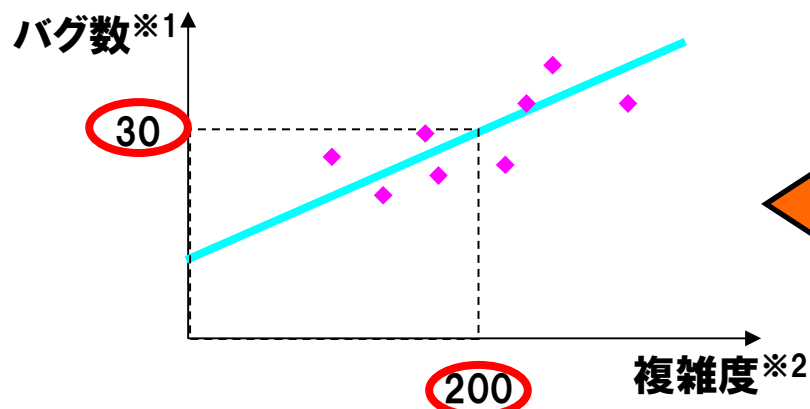
■ 従来の管理方法は、大きく外れることは少なかったが、開発の規模増大、短納期化、差分開発が中心となったことにより、出荷後の品質問題による後戻りが目立ち始めた

予測精度を高める必要性に対処するため  
バグ抽出目標の精度向上にメトリクスを活用

# 事例1「予測精度向上による後戻り防止」

複雑度が高いとバグが多いという傾向をバグ抽出目標の設定に利用して、精度を改善。現在、相関関係を検証中

例：開発中のソースコードの複雑度が200の場合、バグ抽出目標数を30に設定



◆ 複雑度とバグ数の過去の相関データ  
— 相関の式 (近似直線)

適用したシステム (例)	言語	相関係数※
ネットワーク系システム1	C	0.96 (強い)
ネットワーク系システム2	C	0.59 (中程度の強さ)

※相関係数：相関の強さを表す。  
値が1に近いほど相関が強い。

## 算出手順

- 過去のプロジェクト品質データから、複雑度とバグ数の相関の式を回帰分析で算出  
(過去の実績を使うことで、対象ドメインや開発者のスキルの見積への影響を低減)
- バグ抽出目標値を、現在開発中のソースコードの複雑度と上記の相関の式から算出  
(従来はコード規模に計数を掛けて算出)

## 事例2「品質の作り込みによる後戻り防止」

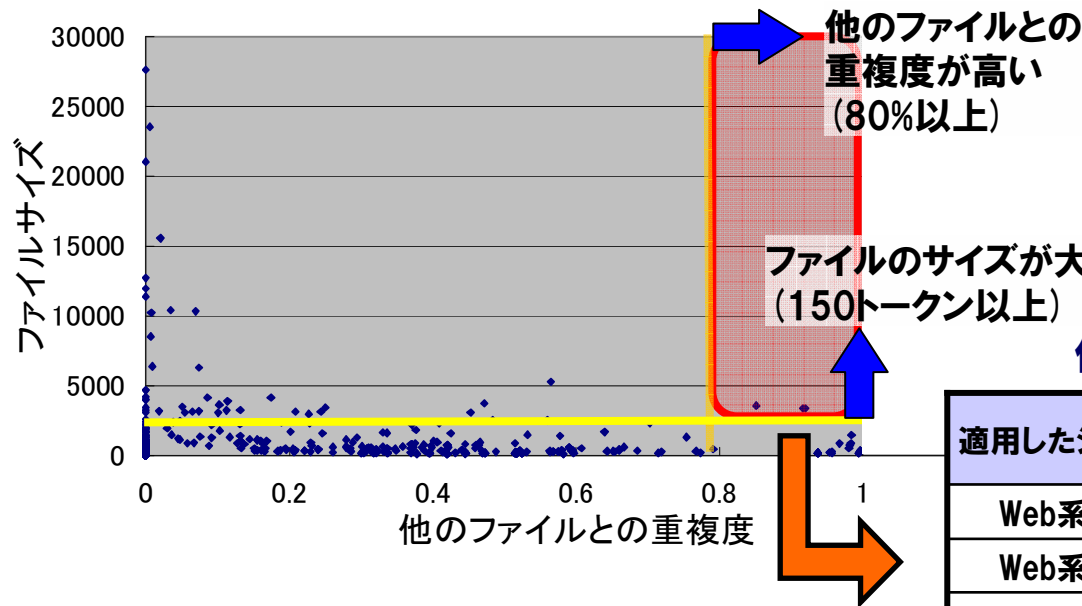
- 保守性品質上、重複コードの抑止が要求事項
- 設計者と実装者が異なることが多く、実装で重複コード発生
- 実装レビューでは、確認対象が多くサンプルリング確認するため、チェック漏れが発生

**コードクローン※のメトリクスを重複コード検出に利用して漏れを防止  
通常、検出ツールは指摘が多数となるため、さらに、絞込み方法を工夫**

※ コードクローン：ソースコードに含まれる同一もしくは類似のコード断片のこと

# 事例2「品質の作り込みによる後戻り防止」

コードクローンメトリクスの重複度とサイズを利用して検出結果を絞り込み  
(絞り込み結果の正解率は80%以上)



他とそっくりなファイルの絞り込み結果

適用したシステム (例)	言語	分析対象ファイル数	絞り込数	絞り込率
Web系システム1	Java	843	14	1.7%
Web系システム2	Java	1,090	37	3.4%
Web系システム3	C#	2,046	37	1.8%

## 他とそっくりなファイルの検出手順

1. コードクローン検出ツールCCFinderX<sup>※1</sup>の大量の分析結果からファイルメトリクスを使い、“他ファイルとの重複度”<sup>※2</sup> (RSA)と“ファイルのトークン数” (LEN) の降順にファイルを並び替え
2. 重複度80%以上、トークン数150以上<sup>※3</sup>のみを抽出

※1: AIST CCFinderX:産業技術総合研究所の神谷年洋氏が開発したツール

※2: 類似した処理を大量に含むファイルの検出では、「ファイル内の類似処理の割合」(RSI) を選別で使用

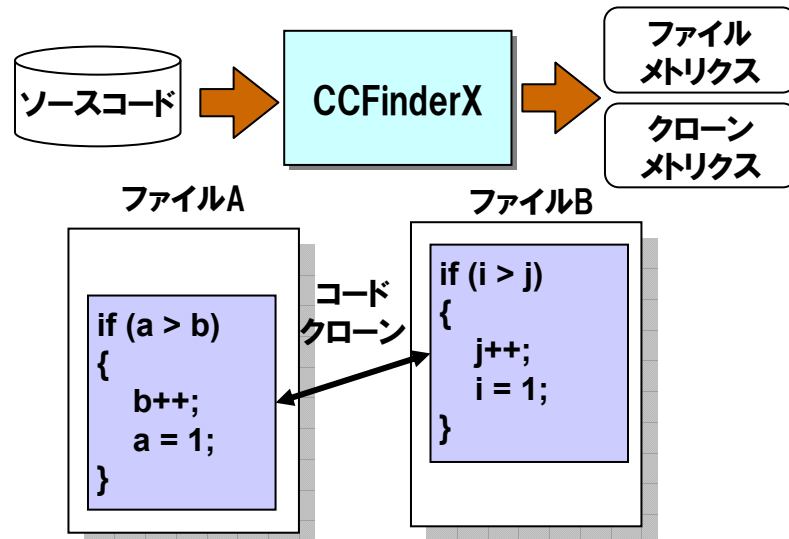
※3: 基準値は、開発現場への適用の経験で得た値

# コードクローン検出ツールCCFinderXとは

- ソースコードに含まれる同一または類似のコードを検出するツール
- ソフトウェア開発の進行状況を定量的に把握するEPMツールに同梱
- NECではCCFinderXを他ツールとの連携容易性に着目して利用

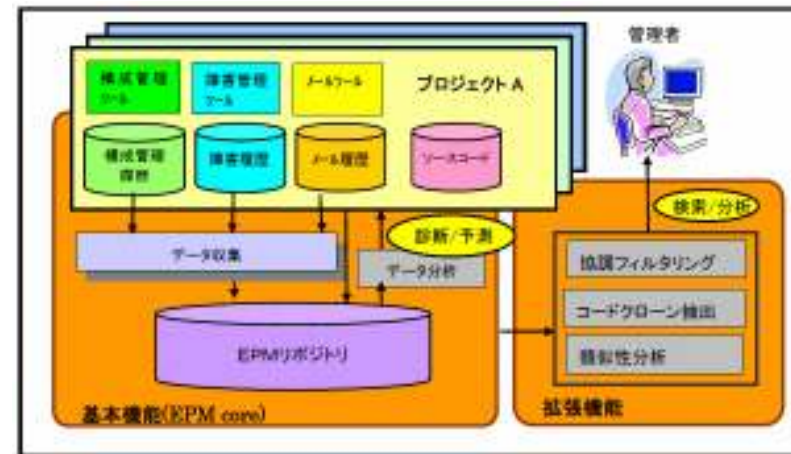
## ■ CCFinderX

- ✓ 産業技術総合研究所 神谷年洋氏が開発
- ✓ ファイルのクローンに関するメトリクス (6種)、クローン自体に関するメトリクス (9種) を測定
- ✓ 対象言語 Java, C/C++, COBOL, VB, C#
- ✓ 登録により無償で利用可能
- ✓ 公式サイト <http://www.ccfinder.net/index-j.html>



## ■ EPMツール

- ✓ 文科省 EASEプロジェクトの開発ツールをIPAが強化
- ✓ 構成管理システムからソースコードの情報、障害追跡システムから障害の情報、メールシステムから開発者の間のメールを収集し、プロジェクトの進行状況を可視化
- ✓ 試用後のアンケートに回答する条件で無償利用可能
- ✓ 公式サイト <http://www.ipa.go.jp/about/toolcatalog/sec/sec-01.html>



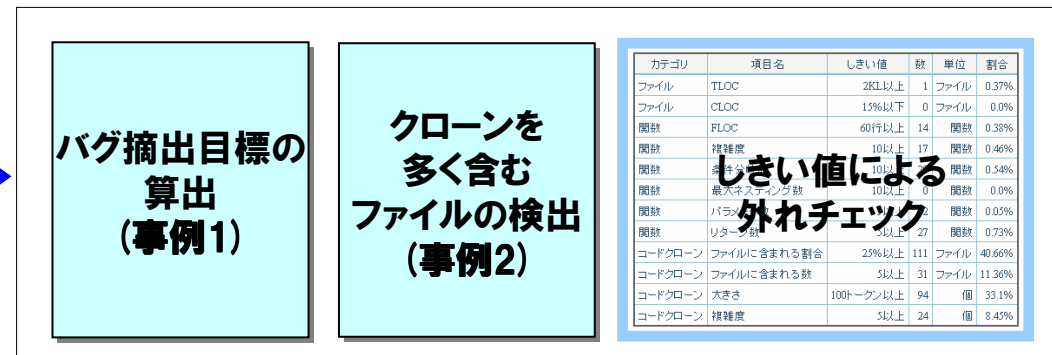
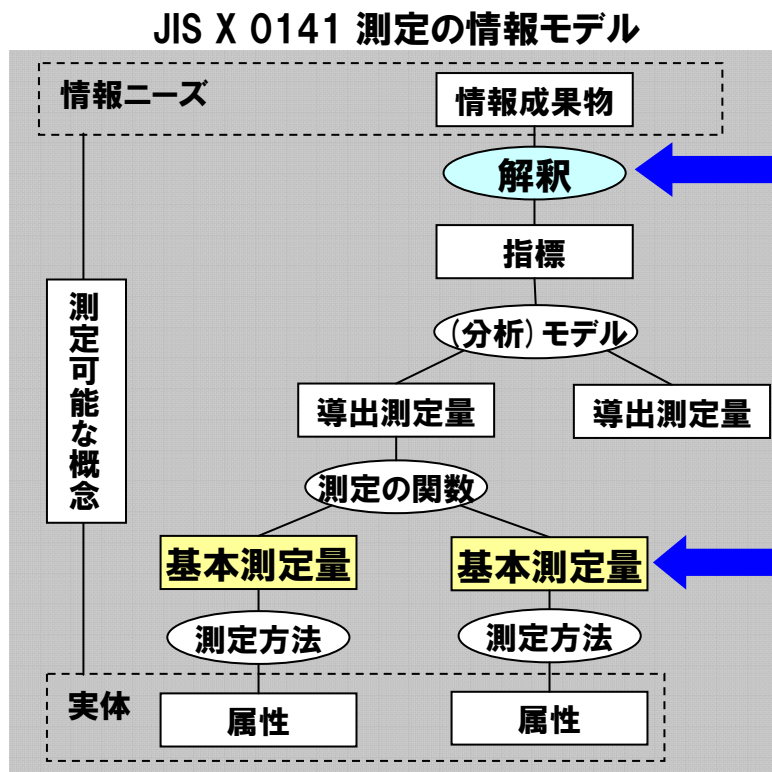
「EASE プロジェクトにおけるツールと分析手法」より引用  
<http://sec.ipa.go.jp/tool/EASE-setsumei.pdf>

# 組織の品質管理のための測定ツールの整備

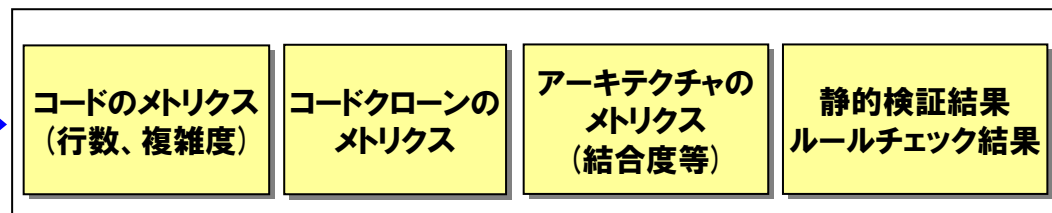
基本測定量の測定方法を統一

品質管理者の管理ノウハウを分析モデルに埋め込み、開発現場でセルフチェックを可能にする仕組みを提供

## ソフトウェア指標分析支援ツールmt

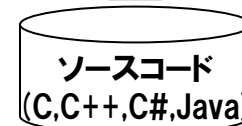


解釈可能な指標の導出



基本測定量の取得

構成管理リポジトリ



# まとめ

---

**大規模、複雑化、短納期化するプロジェクトの品質管理のため、コードメトリクスによる予測精度向上と品質の作り込みで後戻りを防止**

- 複雑度をもとにテストの目標を設定する事例(事例1)  
⇒ **目標値の精度向上による後戻り削減が見込まれる**
- コードクローンの多いファイルを絞り込む事例(事例2)  
⇒ **製造品質向上による後戻り削減が見込まれる**
- 組織としての共通のものさし(ツール)を整備

**現状や改善の効果を客観的に把握できるというメトリクスの特長から、以下の効果を得る**

- 品質管理者の押し付けではない、現場主体の改善の動機付けに繋がる
- 開発者のスキルが底上げされ、結果として生産性が向上

Empowered by Innovation

**NEC**