

# **ソフトウェアメトリクスを用いた 単体テストの品質リスク評価**

**2013年9月13日**

**日本電気株式会社**

**○下村 哲司、森 岳志、佐藤 孝司**

**東洋大学**

**野中 誠**

# 1. はじめに



# 現状の課題と対策

テスト中盤～終盤にかけてのQCD確保が課題

⇒**プロダクトメトリクス**を用いた単体テスト見逃し欠陥リスクの評価と対策

## ソフトウェア開発現場の現状

- ① テスト中盤～終盤に欠陥が流出し、**手戻り工数**が発生
- ② その欠陥の多くは**単体テストで**  
**摘出すべき欠陥**

?

## 本質的な対策

- ① 詳細設計書およびソースコードの**プロダクト品質**の向上
- ② 単体テスト**プロセス品質**の向上

現実的なアプローチ

## プロダクトメトリクスを用いた対策

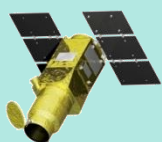
- ① **プロダクトメトリクス**を用いた単体テストの欠陥見逃し**リスクの評価**
- ② 単体テストの欠陥見逃し**リスクに応じた対策**

# 事業領域と主な商品・サービス

パブリック  
向け



セーフティ



衛星システム  
「NEXTAR」



デジタルTV  
送信機

エンター  
プライズ  
向け



各種業種向けソリューション



オフィス向けソリューション

テレコム  
キャリア  
向け



LTE  
ネットワーク製品



海底ケーブル



超小型マイクロ波  
通信システム

スマート  
エネルギー  
向け



xEMS



電気自動車(EV)用  
急速充電器



家庭用蓄電池

ミドルウェア製品の品質保証が責任範囲

CLUSTERPRO SystemDirector

WebOTX

InfoCage

InfoFrame

StarOffice

WebSAM

サーバ



ストレージ



ネットワーク

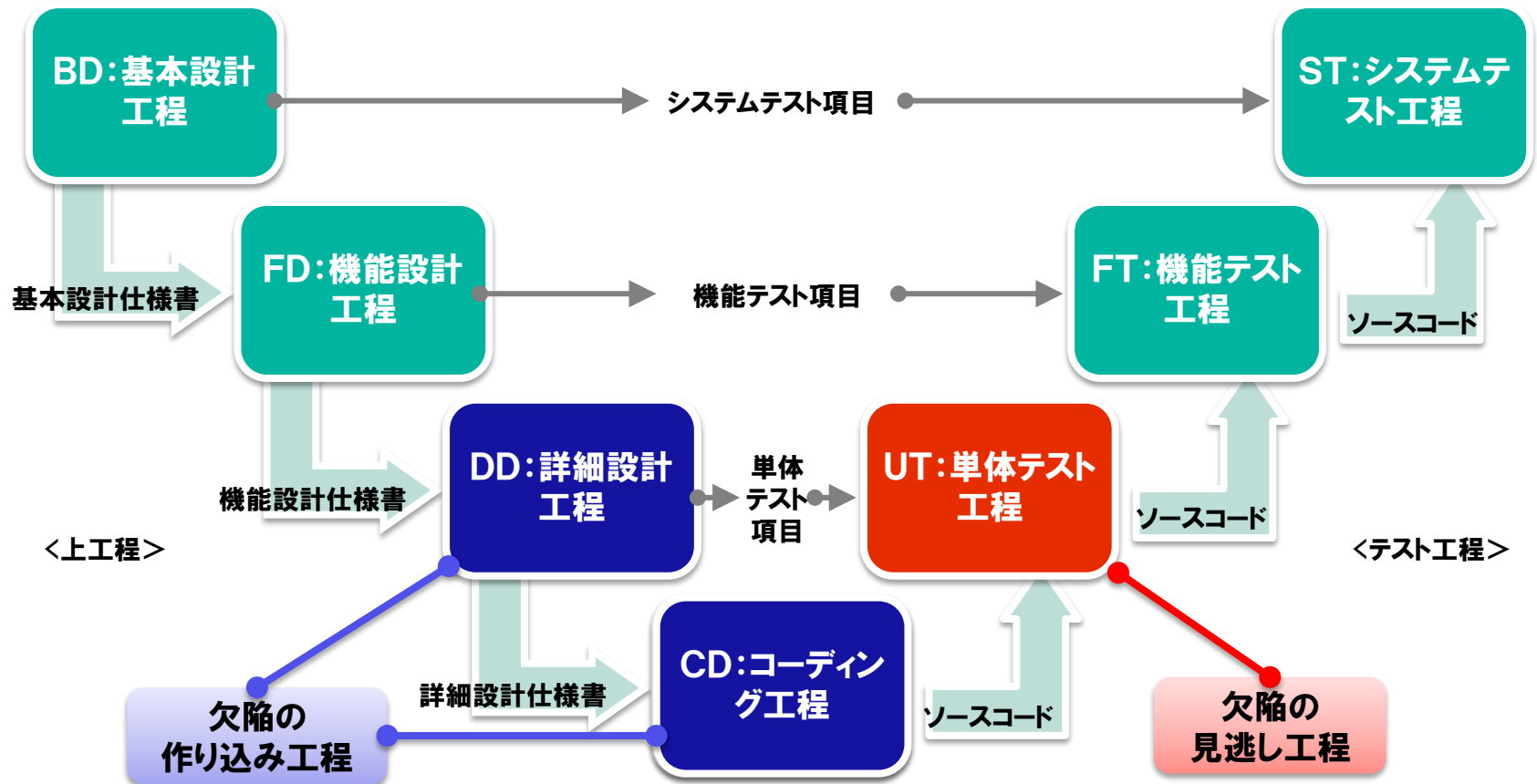


# ソフトウェア開発プロセス

## V字モデルを採用

今回は単体テスト(UT:Unit Test)で摘出すべき欠陥(\*1)に着目

\*1: 詳細設計およびコーディング工程で作り込まれた欠陥

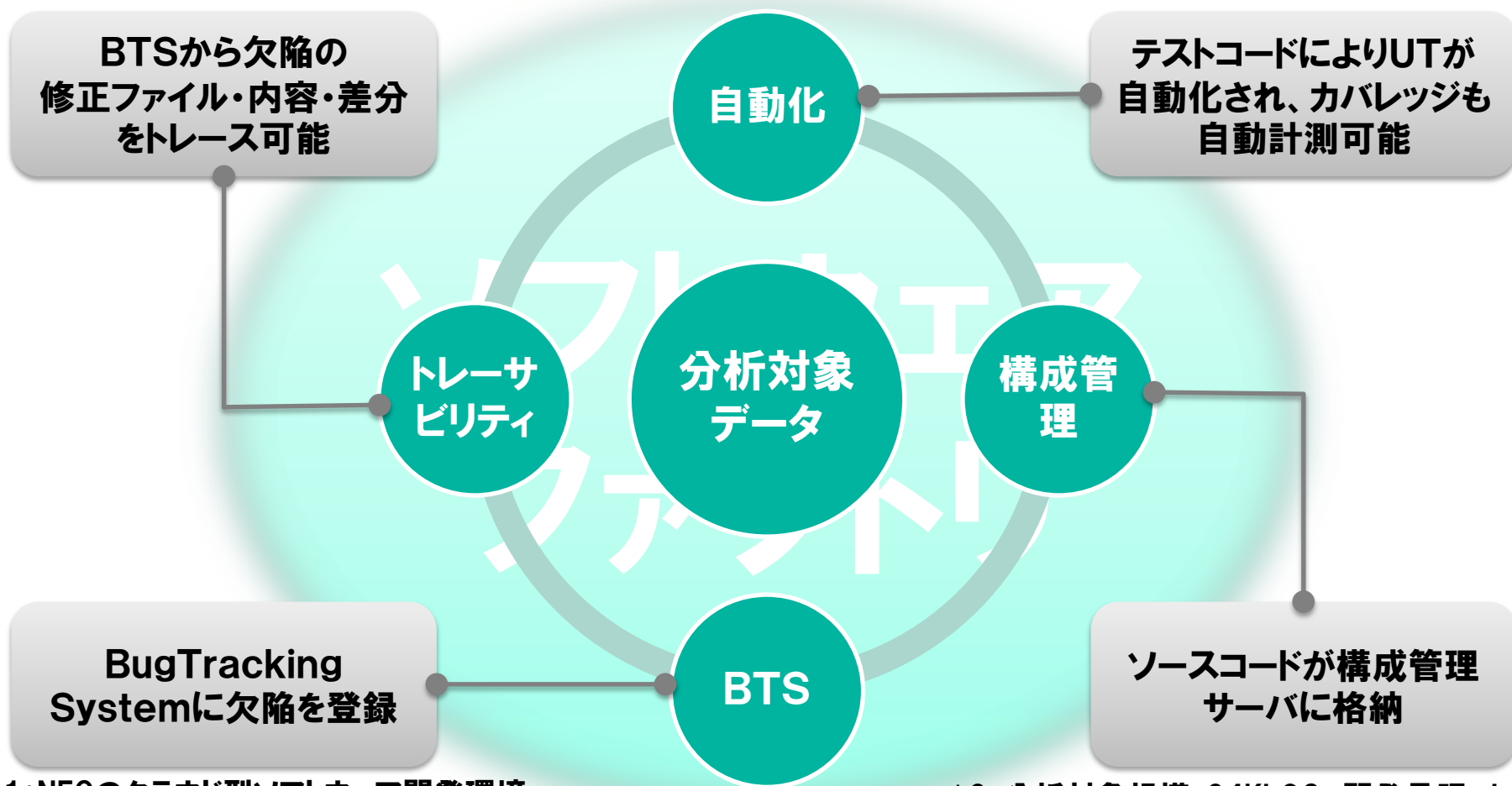


BD:Basic Design, FD:Function Design, DD:Detail Design, CD:CoDing, UT:Unit Test, FT:Function Test, ST:System Test

## 2. 分析対象データとメトリクス

# 分析対象データ

ソフトウェアファクトリ(\*1)を利用している汎用的なミドルウェア製品のうち、以下の条件に合致する製品を分析対象(\*2)とした



\*1: NECのクラウド型ソフトウェア開発環境

\*2: 分析対象規模: 24KLLOC、開発言語: Java

# 分析対象メトリクス

- ・UTの十分性判断尺度として**UT見逃し欠陥数**を採用
- ・ソフトウェアファクトリから**自動収集可能**なソースコードおよびUTメトリクスを採用
- ・各メトリクスは**ファイル単位**(≒Javaのクラス単位)で収集

種別	メトリクス	記号(*1)	説明
欠陥に関するメトリクス	UT見逃し欠陥数	DEF	UTで見逃され、UT完了後～開発完了時までに摘出された欠陥の数
ソースコードメトリクス	規模	ELOC	コメントを除く有効行数
	サイクロマチック数	CYC	McCabeのサイクロマチック複雑度
	分岐条件数	BRA	複合条件を考慮した分岐条件数
	最大ネスティング数	NEST	ファイル内の最大ネスティング数
UTメトリクス	UT項目数	UTI	UT用のテストコードに記述されたテストメソッド数
	UT密度	UTD	UT項目数／KLOC
	C0(*2)	C0	UTのステートメントカバレッジ
	C1(*2)	C1	UTのデシジョンカバレッジ

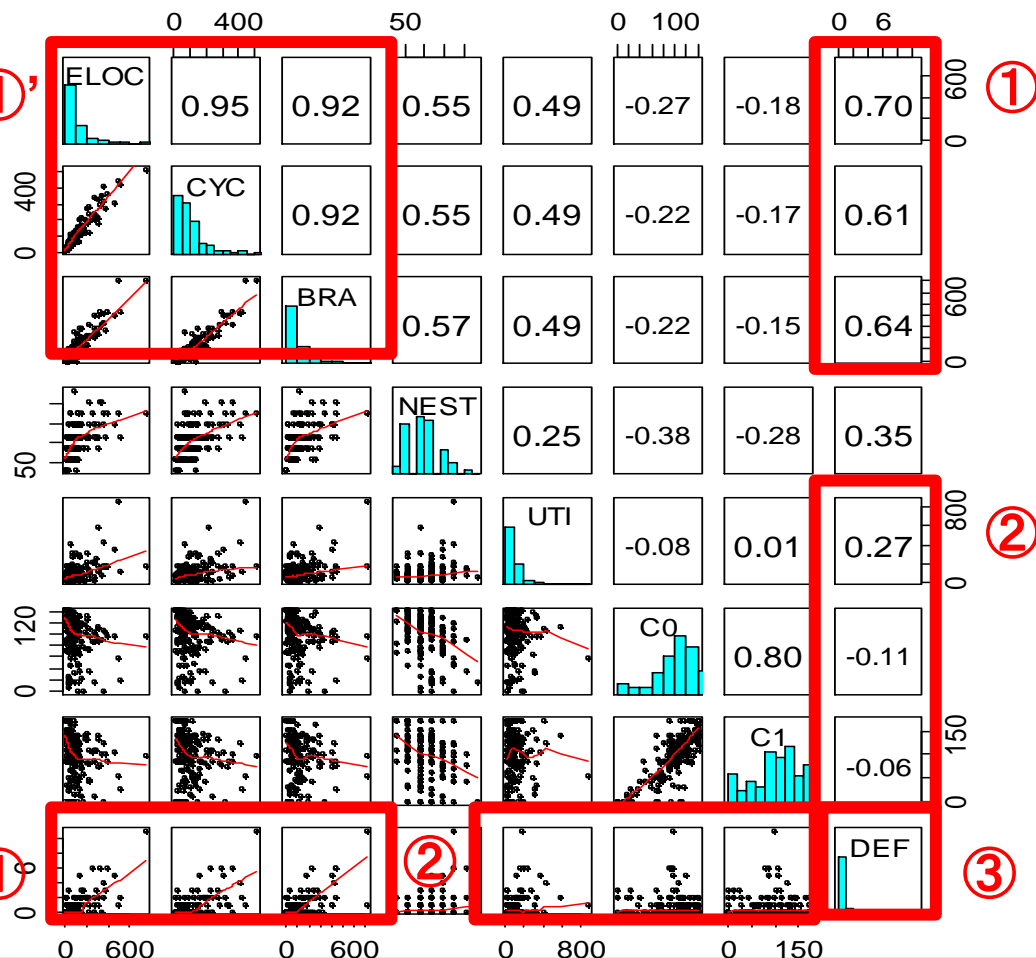
\*1:記号は本スライド中の図中にて使用, \*2:C0とC1をまとめてカバレッジと表現



### 3. 単体テストの品質リスク分析

# メトリクス全体の相関分析

散布図行列を用いて、複数メトリクスの相関を分析した結果、UT見逃し欠陥数は**規模**を用いて説明可能であることが分かった



## 分析結果

- ① UT見逃し欠陥数と、規模/サイクロマチック数/分岐条件数との間に**正の相関**

最も相関の強い**規模**を用いて説明可能

【補足】

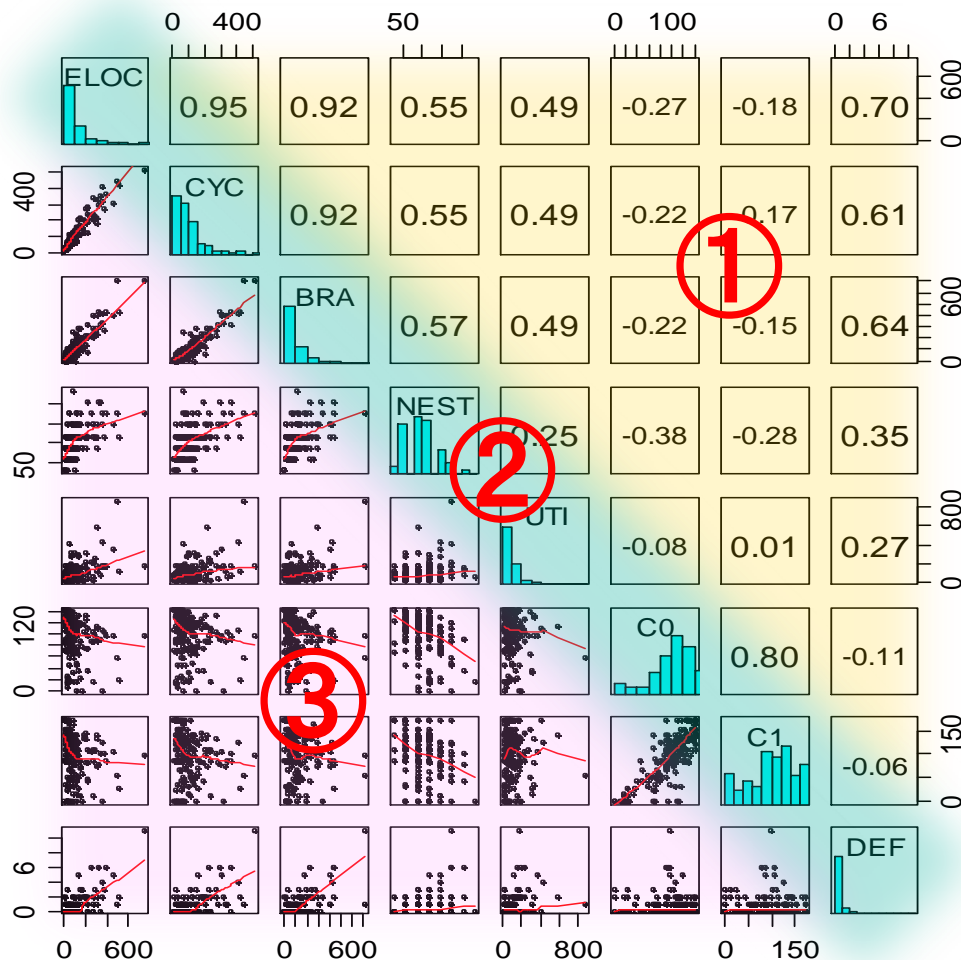
3つのメトリクスには互いに強い相関があり全てを利用する必要はない(①')

- ② UT見逃し欠陥とUTメトリクス(UT項目数、カバレッジ)には**相関なし**
- ③ UT見逃し欠陥が0件のファイルが多い

※目盛は平均値を100とした場合の相対値  
UT見逃し欠陥のみ絶対値

# 【補足】散布図行列

## 散布図行列を用いたメトリクス全体の相関分析



### 散布図行列

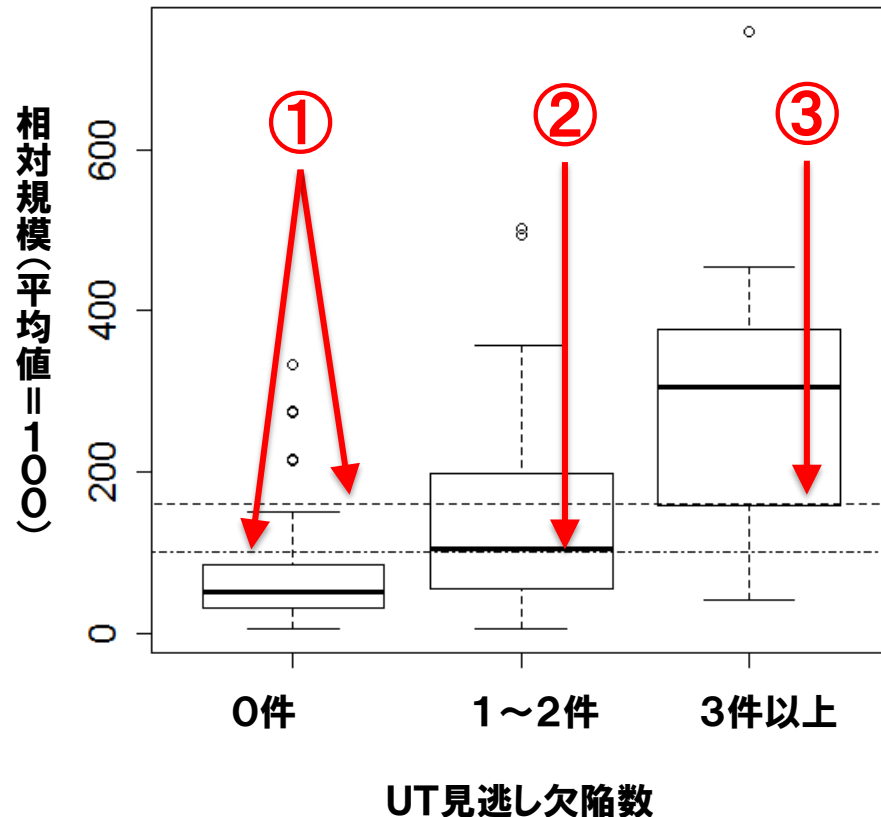
3つ以上のメトリクスを行列形式に配置し、メトリクス間の相関関係を俯瞰する可視化手法

- ① 2つのメトリクスの相関係数の組合せ
- ② 各メトリクスのヒストグラム
- ③ 2つのメトリクスの散布図の組合せ

※目盛は平均値を100とした場合の相対値  
UT見逃し欠陥のみ絶対値

# UT見逃し欠陥の層別分析

UT見逃し欠陥数を層別した結果、**規模の閾値**を導出可能であることが分かった



## 分析結果

規模小のファイルほど欠陥が少なく、規模大のファイルほど欠陥が多い

- ① UT見逃し欠陥数が0件のファイルは、**ほとんどが相対規模160以下に分布し、約75%が相対規模100以下に分布**
- ② UT見逃し欠陥数が1~2件のファイルは、**約50%が相対規模100以上に分布**
- ③ UT見逃し欠陥数が3件以上のファイルは、**約75%が相対規模160以上に分布**

# 分析結果の検証

別の視点(**適合率、再現率、F値**)を用いて、規模の閾値に関して検証を行った結果、UT見逃し欠陥に関する**規模の有効性**を示すことができた

## 規模に閾値を設けた場合の検証結果

閾値	UT見逃し欠陥数	ファイル			欠陥の再現率
		適合率	再現率	F値	
125 <sup>(*)</sup>	>0	72%	58%	64%	74%
	≥3	23%	92%	37%	

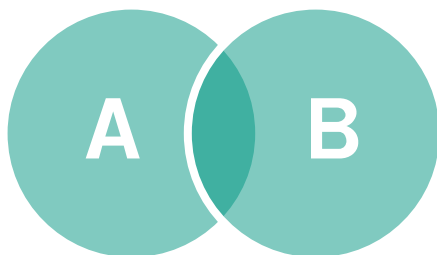
(\*) 相対規模100～160の間で最もF値が高かった値を閾値として評価

- ファイルの適合率より  
閾値外ファイルの**7割強**にUT見逃し欠陥が含まれている
- ファイルの再現率より  
UT見逃し欠陥を含むファイルの**6割弱**、3件以上含むファイルの**9割強**が基準値外のファイルである
- ファイルのF値より  
閾値の有効性が示されている(**64%**)  
※経験的に55%以上有効、65%以上で非常に有効と判断できる
- 欠陥の再現率より  
閾値外ファイルには、全UT見逃し欠陥の**7割強**が含まれている

UT見逃し欠陥  
に関して**規模**  
の有効性を確認

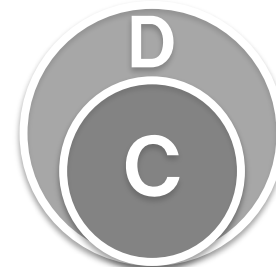
# 【補足】適合率／再現率／F値

ファイル



A: 閾値外ファイル B: 欠陥を含むファイル

UT見逃し欠陥



C: 閾値外ファイルに含まれる欠陥 D: 全欠陥

評価指標項目	式	説明
ファイルの適合率(E)	$(A \cap B) / A$	閾値外ファイルのうち、UT見逃し欠陥を含むファイルの割合
ファイルの再現率(F)	$(A \cap B) / B$	UT見逃し欠陥を含むファイルのうち、閾値外のファイルの割合
ファイルのF値	$(2 \times E \times F) / (E + F)$	ファイルの適合率と再現率の調和平均
欠陥の再現率	$C / D$	全UT見逃し欠陥数のうち閾値外ファイルに含まれるUT見逃し欠陥数の割合

## 4. 単体テストの品質リスク対策

# リスク分析結果のまとめとリスク対策立案方針

## リスク分析結果のまとめ

- ① UT見逃し欠陥数は、**規模**によって説明可能
- ② 規模が一定閾値以上(**規模大**)の場合は、UTで欠陥を見逃す**リスクが高い**
- ③ 規模が一定閾値未満(**規模小**)の場合は、UTで欠陥を見逃す**リスクが低い**
- ④ UT見逃し欠陥の**7割は規模大**、**3割は規模小**のファイルに含まれる

## リスク対策立案方針

- ▶ 特性の異なる各々のリスク(=規模の大／小)に応じた対策が必要
- ▶ 見逃し欠陥の修正内容からリスクに応じた対策を立案



# 見逃し欠陥の修正内容によるリスクに応じた対策

- ・規模大のファイルに対しては仕様ベーステストの強化が有効
- ・規模小のファイルに対しては構造ベーステスト(カバレッジ)の強化が有効

## 欠陥を摘出すべきテスト区分

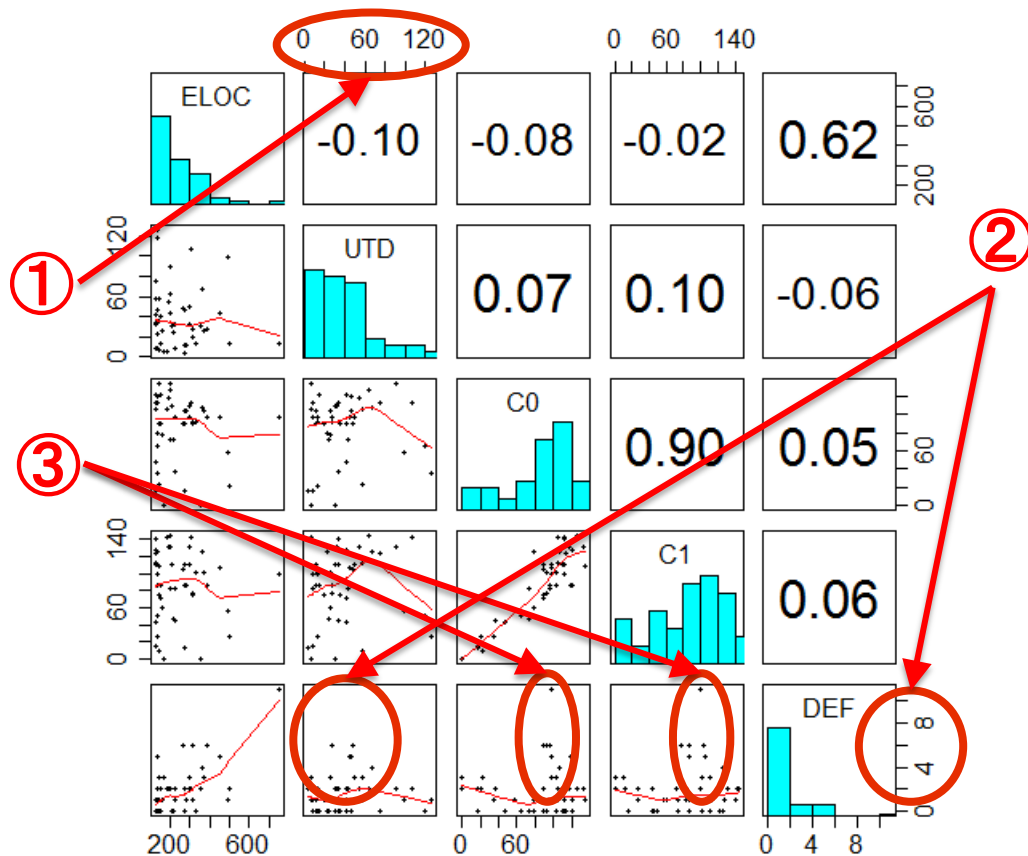
修正ファイルの規模	仕様ベーステストの割合	構造ベーステストの割合
修正ファイルが全て規模「大」の欠陥	78%	22%
修正ファイルが全て規模「小」の欠陥	0%	100%

## テスト区分分類事例

テスト区分	欠陥の原因	実施すべきテスト技法例
構造ベーステスト	クラス名称誤り、設定値誤り	ステートメントカバレッジ
	判定ロジック評価漏れ	デシジョンカバレッジ
仕様ベーステスト	状態遷移評価漏れ	状態遷移テスト
	例外処理評価漏れ	ユースケーステスト
	コマンドオプション評価漏れ	デシジョンテーブル
	入力バリエーション評価漏れ	同値分割／境界値テスト

# 【補足】規模大の場合の散布図行列

規模大の場合は、カバレッジだけでは判断できず、UT密度の底上げが必要  
⇒ **仕様ベーステスト**が不足していることの裏付け



規模大の散布図行列(相対規模 $\geq 125$ )

## 見解

- ① 全体的にUT密度が平均(=100)以下
- ② UT密度が低い場合に欠陥が多い
- ③ C0とC1は相対値100(=平均)近辺でUT見逃し欠陥が多い

## 対策

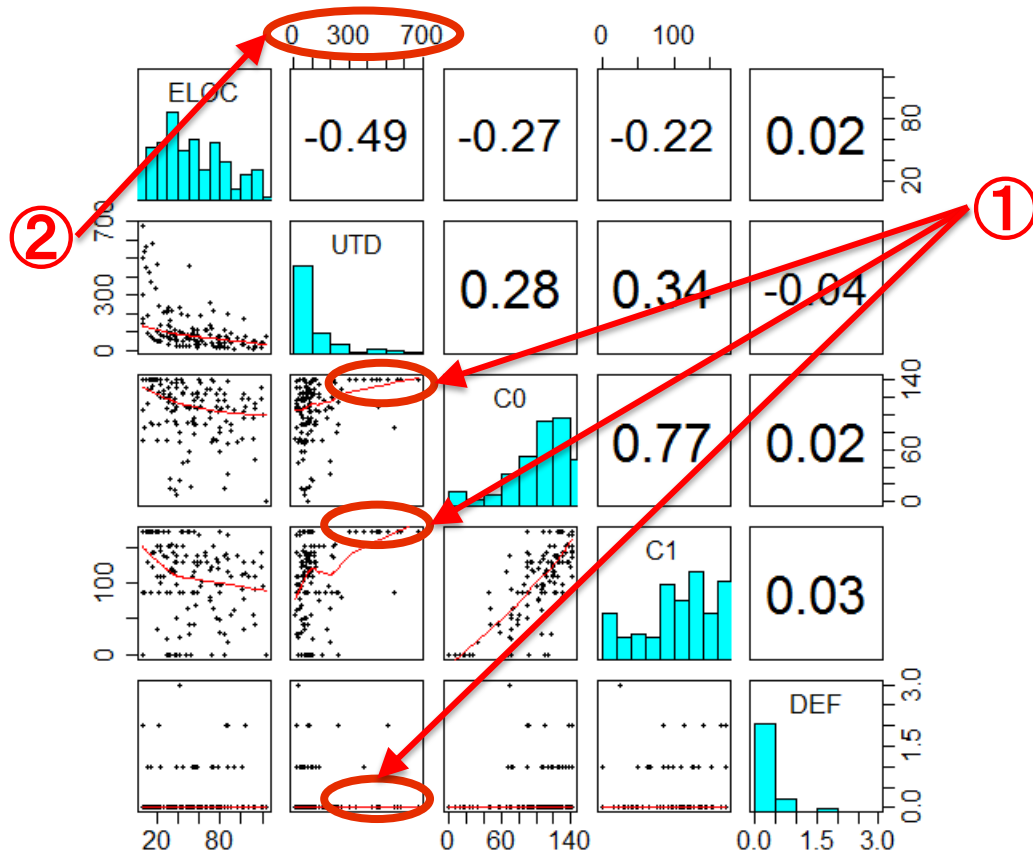
UT見逃し欠陥数を削減する対策

- ▶ UT密度の底上げが必要であり、カバレッジだけでは判断できない

※目盛は平均値を100とした場合の相対値  
UT見逃し欠陥のみ絶対値

# 【補足】規模小の場合の散布図行列

規模小の場合は、カバレッジを強化すればUT見逃し欠陥を削減可能  
⇒ **構造ベーステスト**の必要性の裏付け



規模小の散布図行列(相対規模<125)

## 見解

- ① カバレッジが高くUT密度が高い場合、UT見逃し欠陥数が少ない
- ② 規模大と比較して、全体的にUT密度が高い

## 対策

UT見逃し欠陥数を削減する対策  
▶ カバレッジ強化

※目盛は平均値を100とした場合の相対値  
UT見逃し欠陥のみ絶対値

# 5. おわりに



# メトリクスを用いた先行研究事例

- ・実製品データに適用した事例は多くはない
  - ・テスト項目数やカバレッジとプロダクトメトリクスを組み合わせた品質判断の必要性あり
- ⇒ **本論文はこれらの課題を満足**

サイクロマチック数を用いたテスト方法論(\*1)  
(サイクロマチック数=テスト項目数)  
⇒実製品に対する評価の必要あり

Rosenberg, Stapko, Galloの提案(\*2)  
Lorenzの経験則(\*3)  
⇒弊社実製品に対しては情報量が不足(\*4)

## 先行研究事例

ソフトウェア設計品質の定量的評価(\*4)  
⇒テストメトリクス(テスト項目数/カバレッジ)  
との関係には言及していない

Fault-Proneクラス予測精度向上(\*5)  
メトリクスの効果を活かす利用方法(\*6)  
⇒テスト項目数/カバレッジと組み合わせた  
品質判断の必要性を示唆

(\*1) Watson, A. and McCabe, T. Structured Testing : A Testing Methodology Using the Cyclomatic Complexity Metric, NIST Special Publication , Vol.500-235, 1996.

(\*2) Rosenberg, L., Stapko, R., and Gallo, A., Applying Object-Oriented Metrics, 6th Int'l Symposium on Software Metrics, Nov. 1999.

(\*3) Lorenz, M. and Kidd, J., Object-Oriented Software Metrics: A Practical Guide, PTR Prentice Hall, 1994.

(\*4) 倉下亮, 吉村博昭, 野中誠, 菅田直美 : CKメトリクスの分布に基づくソフトウェア設計の質の定量的評価, ソフトウェア品質シンポジウム2011報文集, 2011.

(\*5) 額根信子, 川村真弥, 野村准一, 野中誠 : プロセスメトリクス利用による Fault-Prone クラス予測精度の向上, ソフトウェア品質シンポジウム2010報文集, 2010.

(\*6) 額根信子 : メトリクスの効果を生かすための利用方法とは?, ソフトウェア品質シンポジウム2011招待講演, 2011.

# まとめと考察

## UTの十分性に影響を与えるメトリクス

- **規模**が最も有用なメトリクス
- 【規模が一定以上の場合】 UTの品質リスクが**高い**
- 【規模が一定未満の場合】 UTの品質リスクが**低い**

## UTの十分性の判断

- 単純にカバレッジだけでは判断できない
- 【規模が一定以上の場合】 **仕様ベースドテスト**が十分性に影響あり
- 【規模が一定未満の場合】 **カバレッジ**を用いて十分性を判断できる可能性あり

## UTの十分性を確保する対策

- 単体テスト対象規模を可能な範囲でコントロール
- 【規模が一定以上の場合】 **仕様ベースドテスト強化**
- 【規模が一定未満の場合】 **カバレッジ強化**

# 今後の展開



- ・規模の閾値によるリスク特定と対策を適用可能な範囲で実施

- ・ソフトウェアファクトリ利用拡大に伴いUT自動化製品が増加
- ・今回と同じ粒度でメトリクス測定が可能な製品が増加

- ・組織全体の基準値および対策方針の立案と施行
- ・製品特性の考慮も必要(開発言語、事業領域、フレームワークなど)

- ・モジュールのコンパクト化によりインタフェースの重要性アップ
- ・モジュール間のメトリクスを用いた分析と対策の立案

# 参考

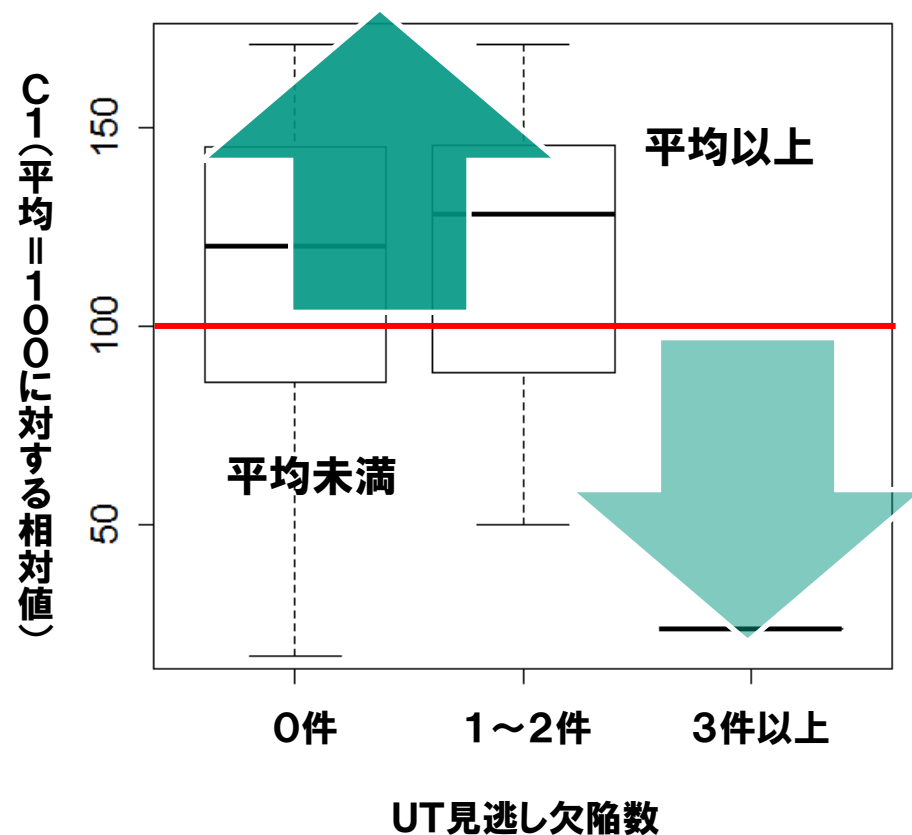
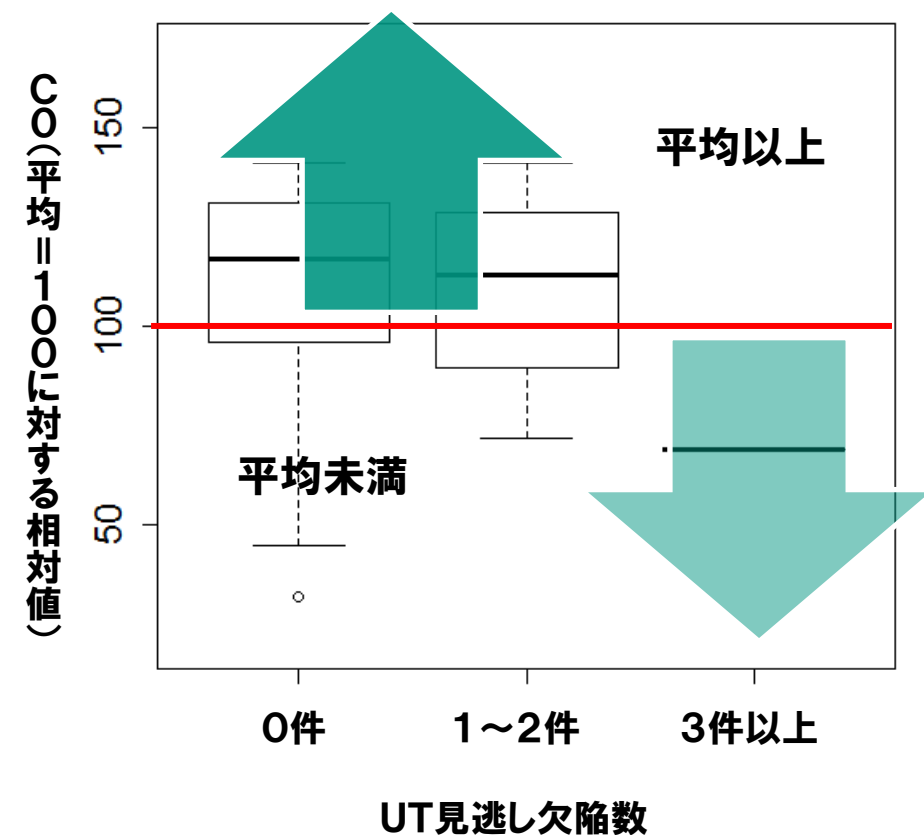




## 【参考】規模小の対策の補足説明～見逃し欠陥を層別した場合のカバレッジ

規模小の場合、UT見逃し欠陥数が多いとカバレッジが低い  
⇒ **カバレッジ強化**の必要性を示唆

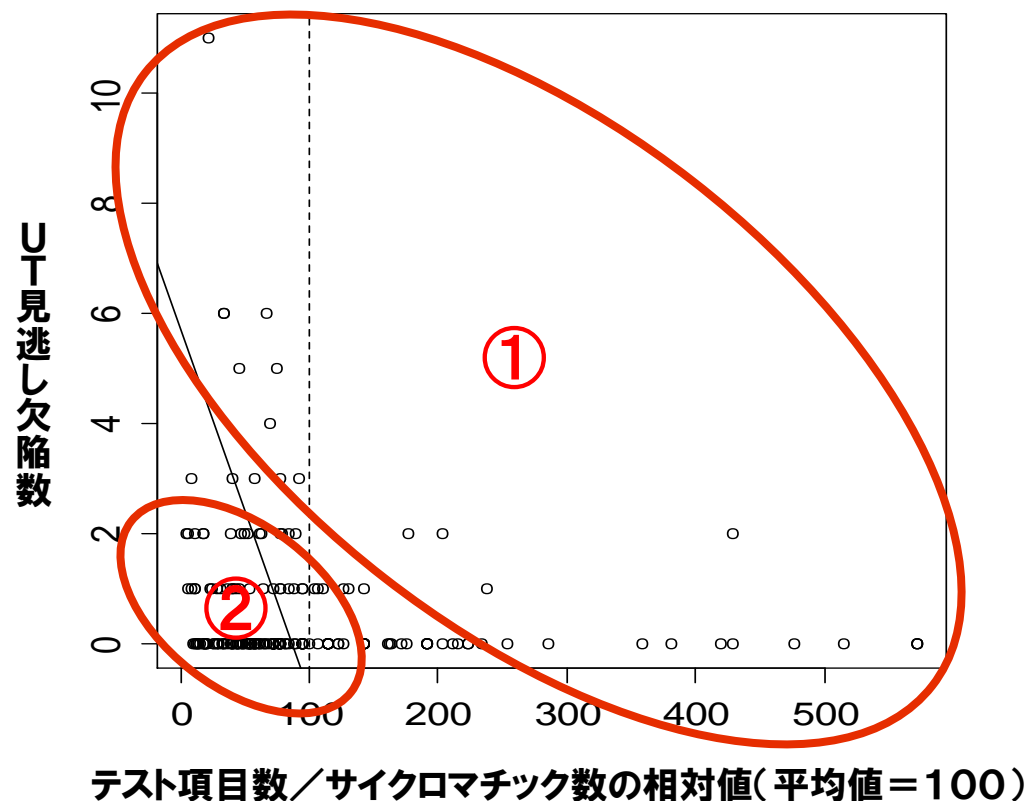
### 規模小の場合の層別したUT見逃し欠陥数に対するカバレッジ分布



## 【参考】先行研究に対する補足説明～サイクロマチック数とテスト項目数～

実製品に対して、単純にサイクロマチック数とテスト項目数を比較するだけでは、単体テストの十分性を評価することは難しい

### サイクロマチック数あたりのテスト項目数とUT見逃し欠陥数の散布図



#### 仮説

単体テスト／サイクロマチック数が

- ① 大きい場合 ⇒ 品質良い
- ② 小さい場合 ⇒ 品質悪い

#### 分析結果

- ① 概ね仮説通り
- ② 説明できない場合が多い

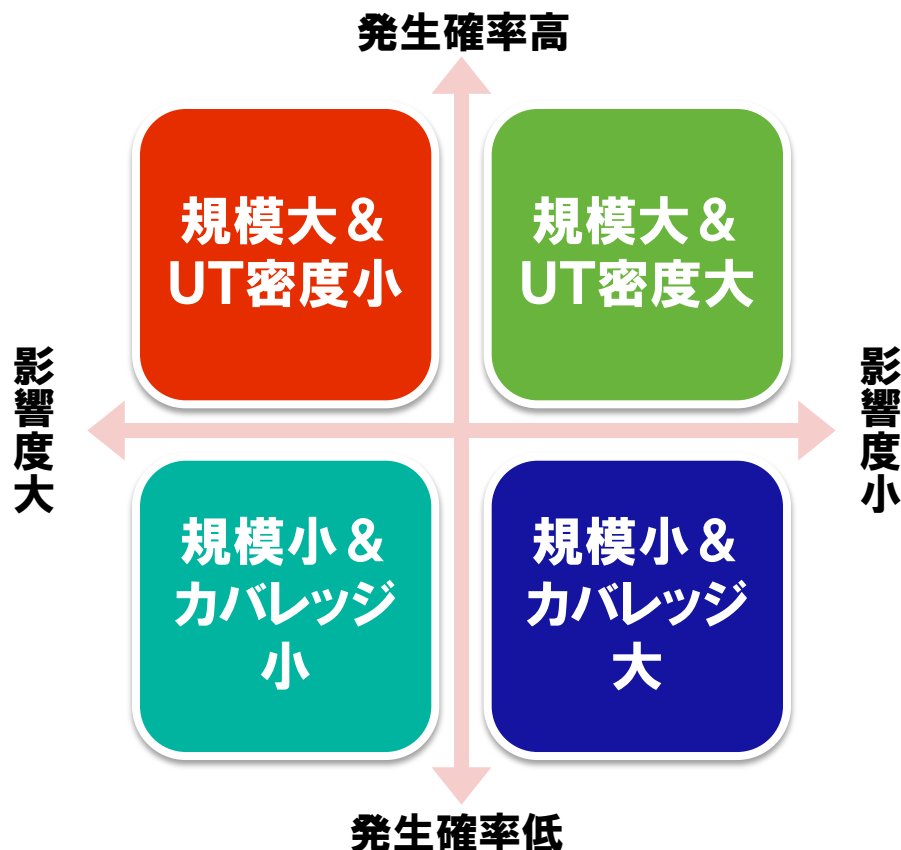
単純にテスト項目数の基準値をサイクロマチック数に等しくすることは難しい

# 【参考】まとめと考察の補足説明～定性的リスク分析への活用～

規模の大小によるリスク対策を定性的リスク分析として発生確率・影響度マトリクスにマッピング

## 発生確率・影響度マトリクス

- (※1) 発生確率: 見逃し欠陥がUT後に摘出される確率  
(※2) 影響度: リスク発生時に対策を実施する場合の手戻り工数



## 活用方法例

発生確率・影響度マトリクスを組織別・製品別の優先度を割り当ててリスク管理を実践する

### ① 規模大 & UT密度小の予防策

UT密度の底上げ施策として、**仕様ベーステスト**の重点強化

### ② 規模小 & カバレッジ小の予防策

**構造ベーステスト(カバレッジ)**の強化

NECグループビジョン2017

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



Empowered by Innovation

**NEC**