

静的解析結果の見せる化

～並べて、比べて、区分けして、気づきを促す～

2015年10月21日

キヤノンITソリューションズ株式会社

品質監理本部 品質監理部

和良品文之丞

Canon

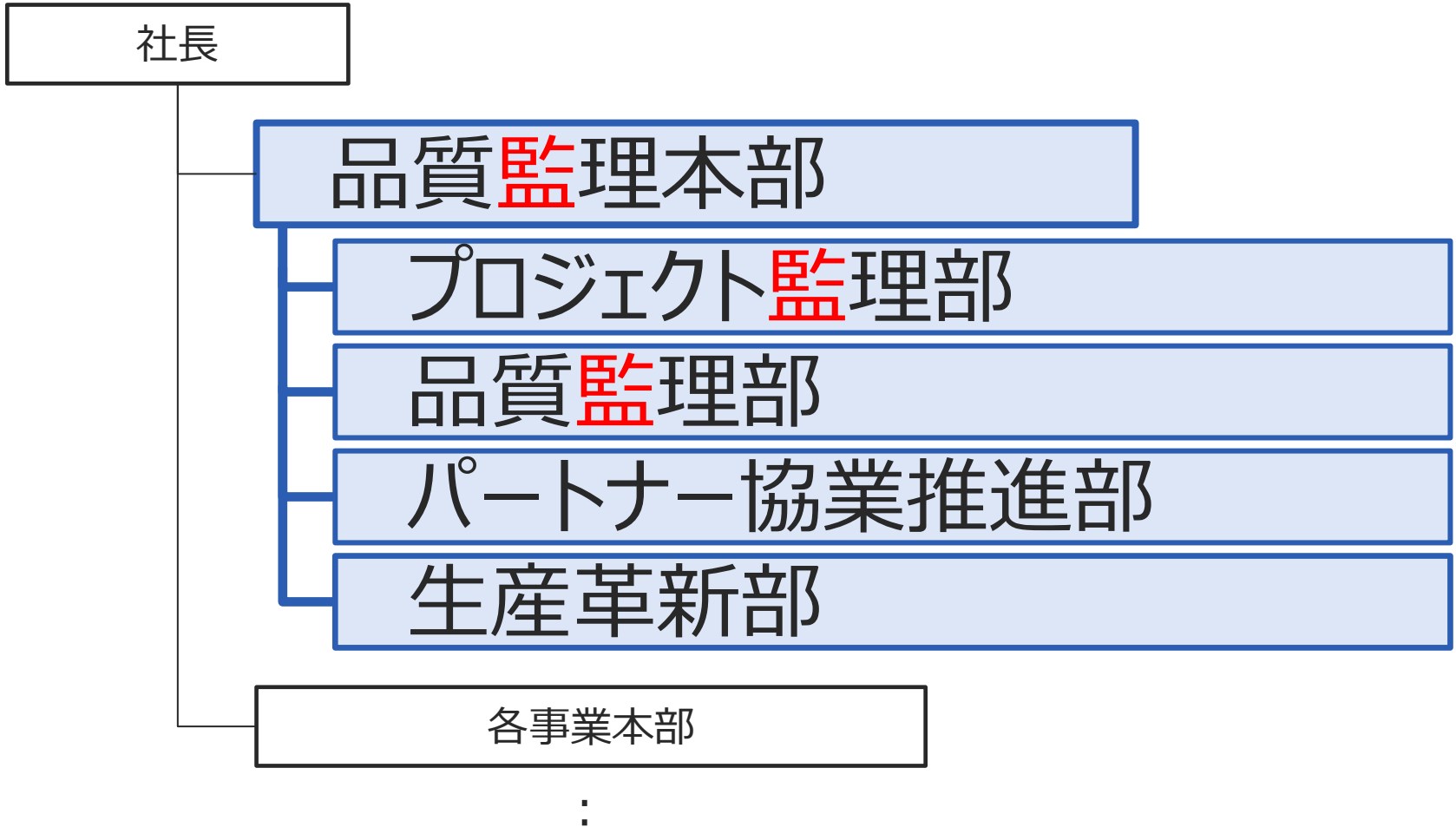
目次

1. 品質監理本部紹介
2. 静的解析ツール導入の経緯と課題
3. 課題の分析と対応策
4. 見える化の実現
5. 改善効果
6. 改善活動の評価と今後の課題

1.1. キヤノンITソリューションズ（株）概要

主要事業	SIおよびコンサルティング、各種ソフトウェアの開発・販売
設立	1982年7月1日
所在地	東京都品川区東品川2-4-11 野村不動産天王洲ビル
代表者	代表取締役社長 神森 晶久
資本金	36億1,700万円
従業員数	3,542人（2014年12月末日現在 連結）
売上高	869億2,200万円（2014年12月期 連結）

1.2. 「品質監理本部」の組織構成



1.3. 「品質監理本部」のミッション

- 品質・生産性を向上させるための、「しくみ・しかけ」をマネジメント面、エンジニアリング面にわたり、ブラッシュアップしながら継続して、提供する
- ノウハウの整理・情報発信を行い、社内外の好事例を紹介し、**現場にはいって献身的に支援**することで、社内への普及・定着を実現する
- これらの活動について、社内外に情報発信していく

献身的な「黒子」であれ

1.4. 特徴1 「管理」でなく「監理」に込めた思い

「監」

● 語源 (※諸説あります)

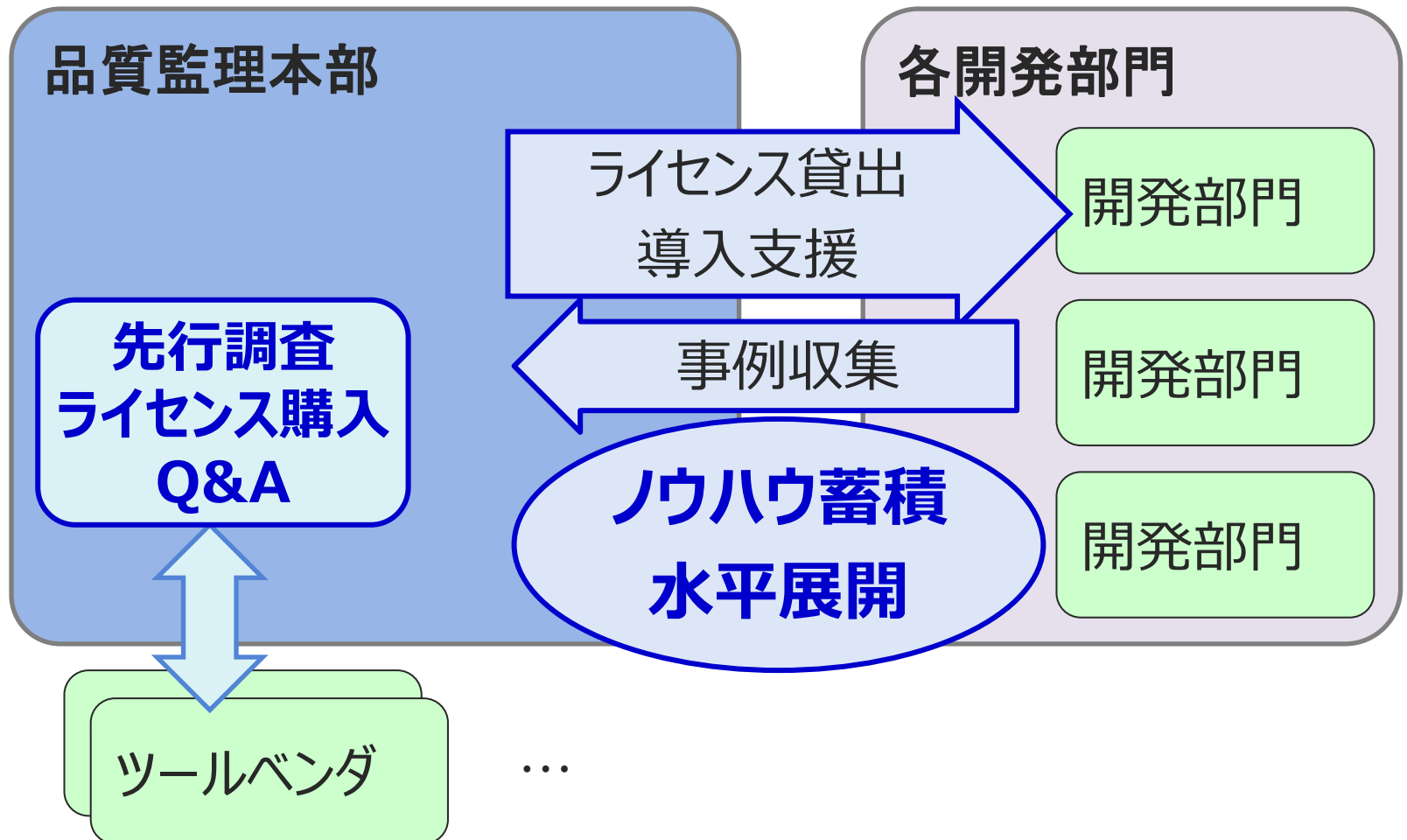
下の“皿”という部分が、目を横にし体を横にして「天井から覗き込む」、「やさしくみつめる」という意味を持つとのこと。

● 解釈

厳しく「管理」するのでなく、寄り添いながら「監理」していく

1.5. 特徴2 ツール活用ノウハウの蓄積と展開

- 全社でノウハウを蓄積、開発部門へ横展開



1.6. 特徴3 積極的な情報共有の推進

● 品監本部の情報発信の代表例

社内：オープンセミナー

- 第33回 2月 クラウド特集
- 第34回 2月 オフショア・ニアショア特集
- 第35回 4月 アジャイル開発特集
- 第36回 5月 製品・サービス紹介
- 第37回 6月 GUIテスト自動化の事例紹介
- 第38回 9月 アジャイル開発特集 第2弾

社外：Tech & Quality Report

- 第5号 1月発行 2015年のITナビゲータ特集
- 第6号 4月発行 デジタルマーケティング特集
- 第7号 6月発行 キヤノンITS認定プロジェクト
マネージャ特集
- 第8号 9月発行 IoT/M2M特集



目次

1. 品質監理本部紹介
- 2. 静的解析ツール導入の経緯と課題**
3. 課題の分析と対応策
4. 見える化の実現
5. 改善効果
6. 改善活動の評価と今後の課題

2.1. 品質向上活動でツール導入を推進

プロジェクト管理

大型プロジェクトの失敗撲滅

+

プロセス管理

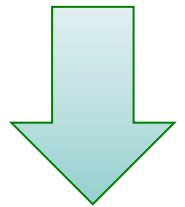
事業別QMSとノウハウの継承

+

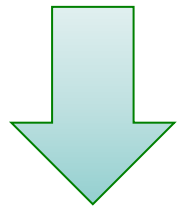
プロダクト品質管理

- ・測定・分析プロセスの導入
- ・ツール導入の方針策定

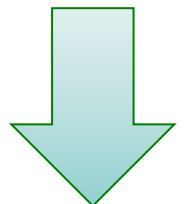
2007



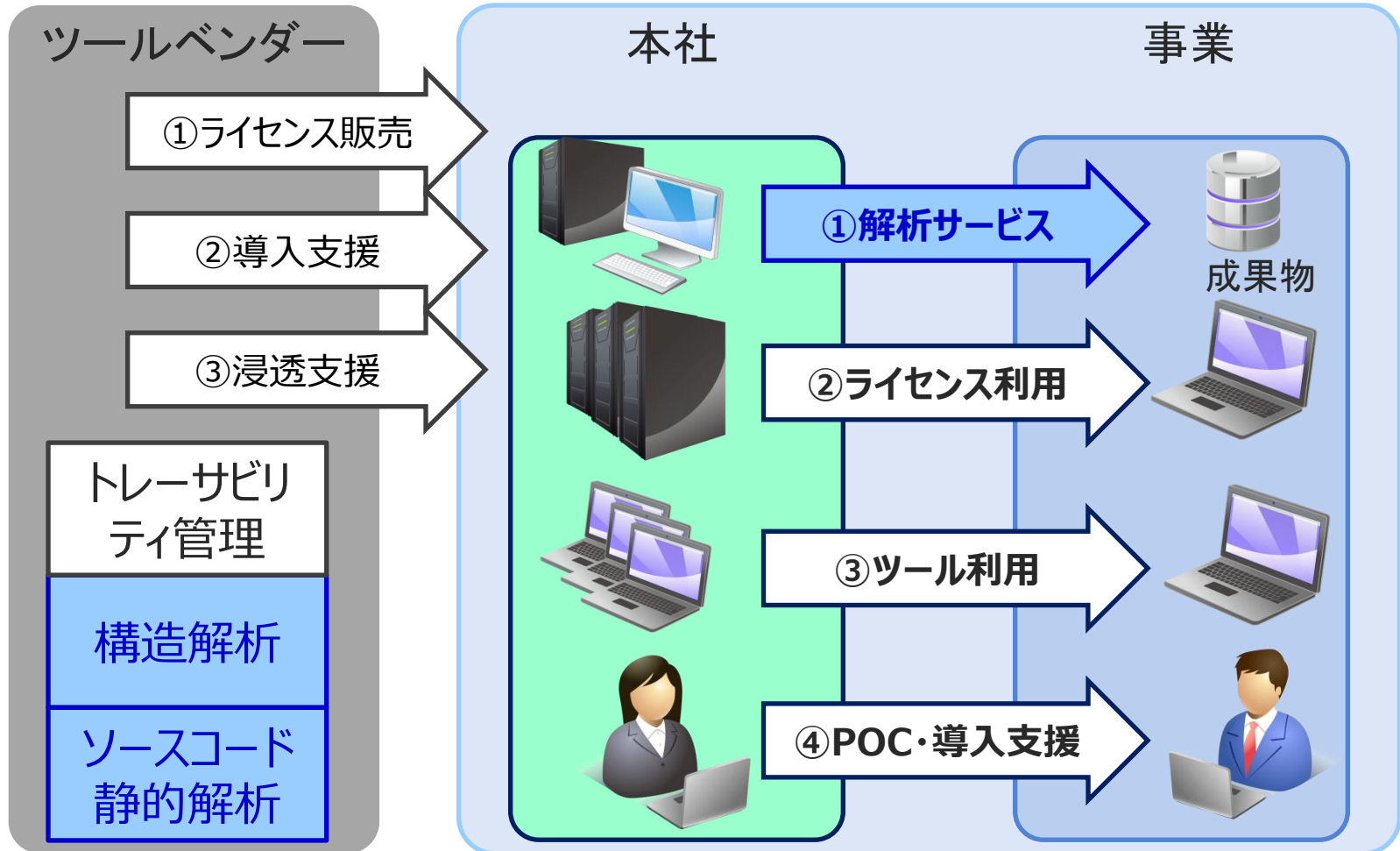
2011



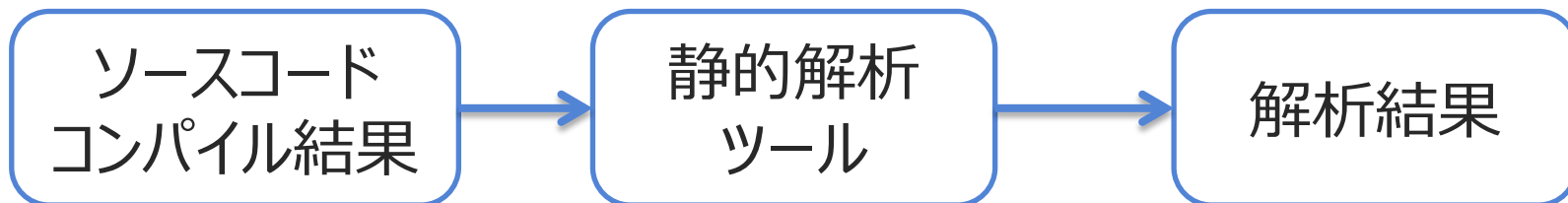
2013



2.2. 普及を意図して解析サービスを実施

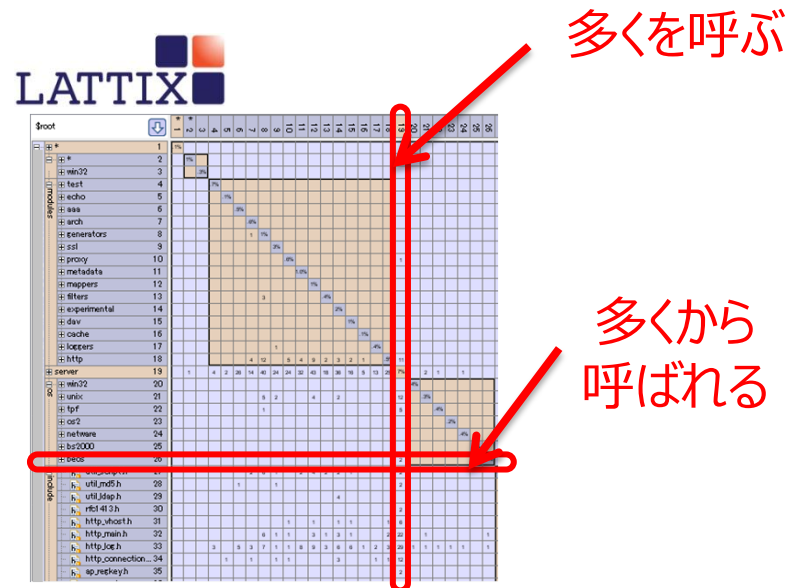


2.3. 解析結果報告の例



マトリクスツリーマップ

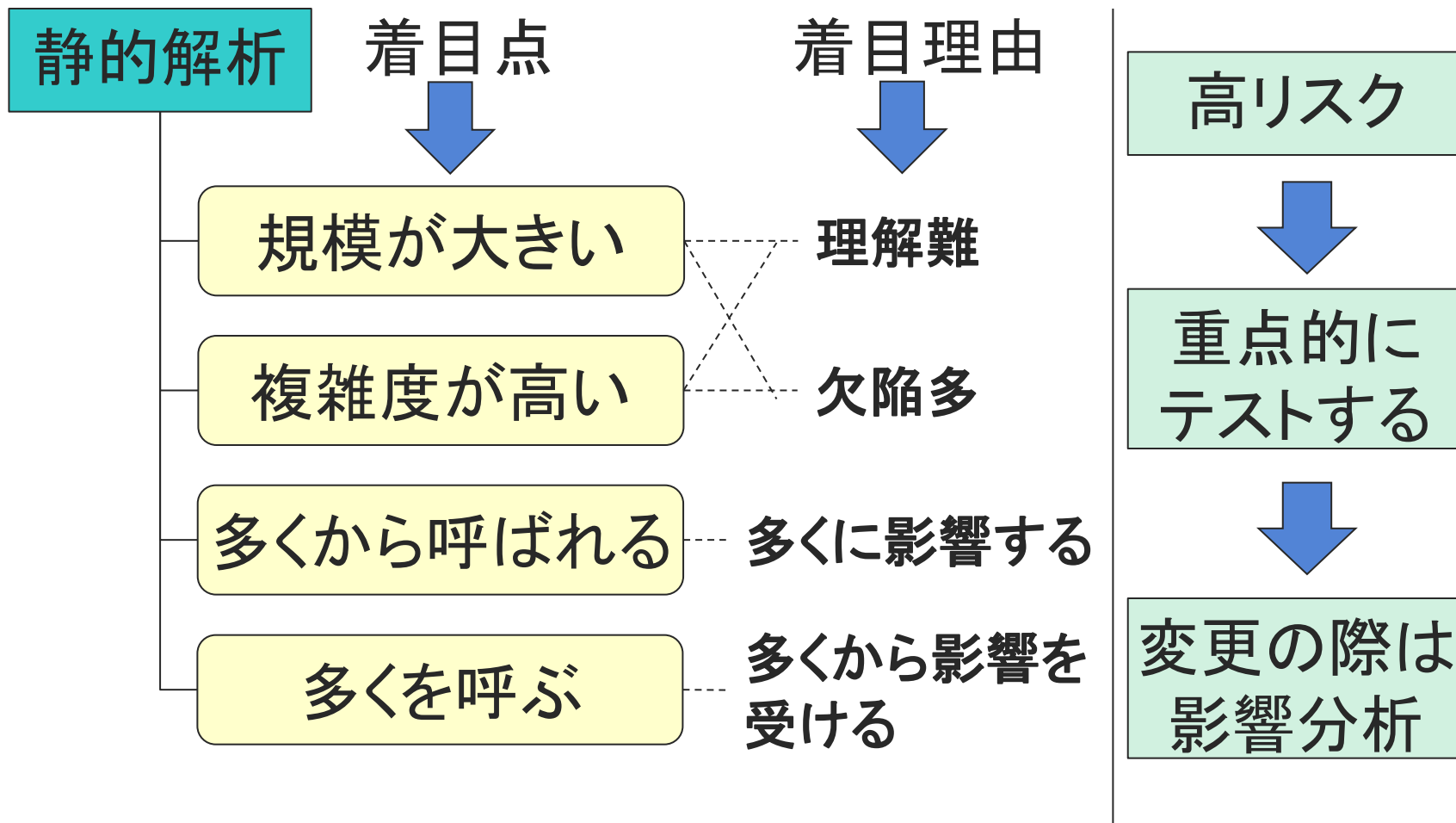
- ・箱の大きさ: プログラムの規模
- ・色の濃さ: プログラムの複雑さ



ディペンデンスストラクチャーマトリックス

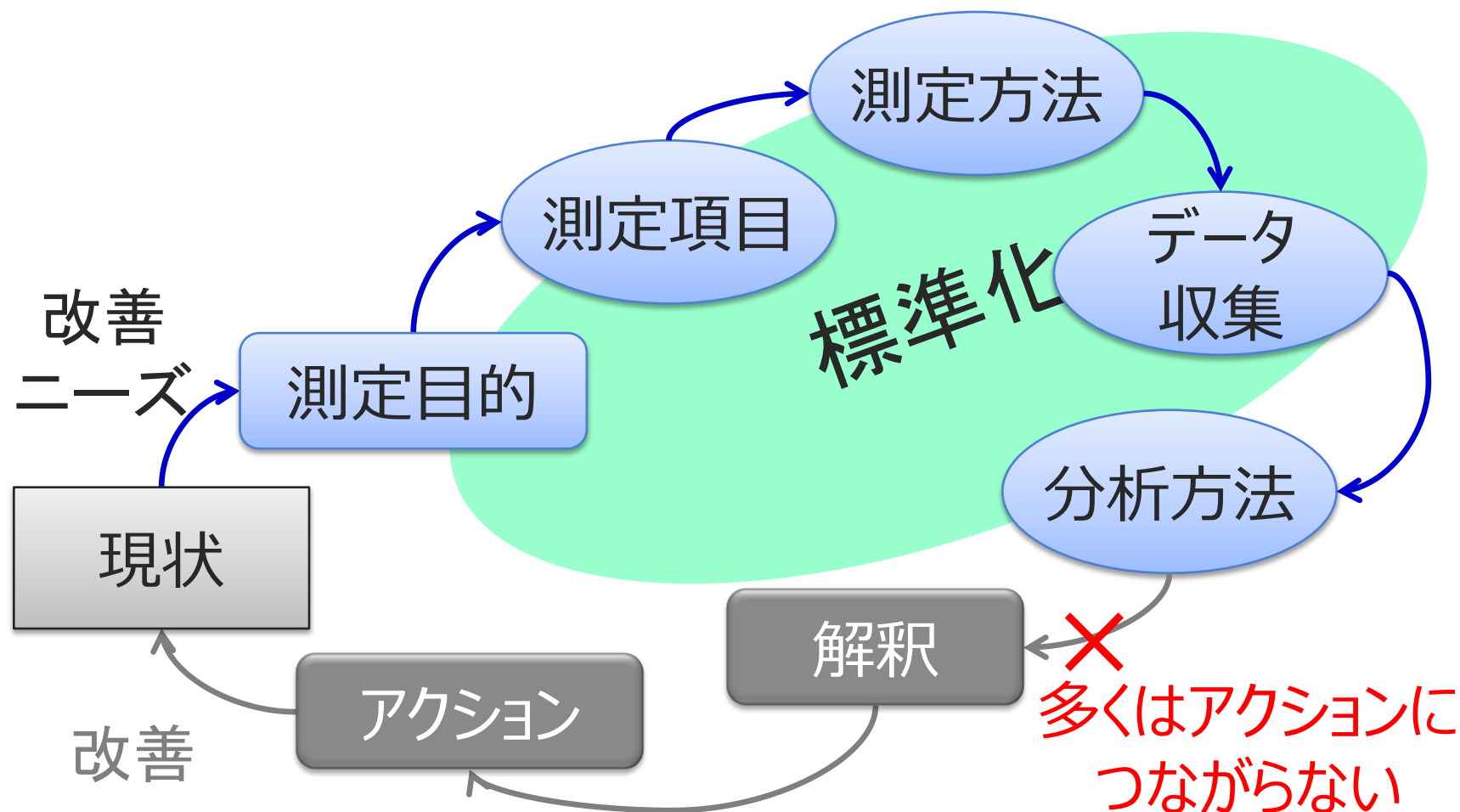
- ・縦の列: 何を参照しているか
- ・横の行: 何から参照されているか

2.4. 訴求のポイント：品質リスク



2.5. 課題

- 一部の部署ではリファクタリング、重点テストを実施したが・・・



目次

1. 品質監理本部紹介
2. 静的解析ツール導入の経緯と課題
- 3. 課題の分析と対応策**
4. 見える化の実現
5. 改善効果
6. 改善活動の評価と今後の課題

3.1. 現場の意見

どこが複雑で規模が大きいかは、開発者であれば改めて言われるまでもなくわかっている。

示された状況からは、それが良いのか、悪いのか、よくわからない。

今稼動しているものを変えるのは困難で、計画的にリファクタリングを組み入れるのであれば、判断基準が必要である。



何をしたら良いか、わからない
ツールのメリットも伝わらない



3.2. 原因分析（追加ヒアリングの結果）

情報量が乏しい

複雑度、規模の高いもの 1 ~ 2 個を示す
→ 開発者が感覚的に判断できる
→ ツールを使うメリットを訴求できていない

プログラム単体の
報告

モジュール間の比較、コード行の種類のカテゴリ
→ 解析対象のプログラムで閉じていた
→ ベンチマークのような評価ができていない

単なる事実の
報告

静的解析結果を、わかりやすく図示したつもり
→ 値が示す意味や解釈に踏み込めていない
→ 良し悪しがわからず、アクションにつながらず



見せ方をもっと工夫しよう
【見せる化】

3.3. 見せる化：見せ方の工夫（案）

並べる

複雑度、規模の高いもの 1 ~ 2 個を示す
→絶対値による順位を複数個、並べて示す
(メトリクスツリーマップの相対値比較を補完)

比べる

モジュール間の比較、コード行の種類のカテゴリ
→規模の大きなOSSのプログラムと比較する
リファクタリングの前後 (Before/After) を比べる

区別
する

静的解析結果を、わかりやすく図示したつもり
→文献から複雑度と規模にそれぞれしきい値を設定
双方がしきい値よりも大：危険ゾーンで示す

3.4. しきい値の設定例

	説明	しきい値 (例)	説明
①	メソッド行数	≥ 100	大きな関数
②	サイクロマティック複雑度	> 20	複雑
		> 50	メンテ困難、リファクタリングが要
③	本質的複雑度	> 10	複雑な構造
④	保守難度	サイクロマティック複雑度 ≥ 10 & $((③/②) \geq 0.5)$	コード保守が難しい
⑤	クラス結合度	≥ 30	クラス間の依存が高く、保守が難しい
⑥	クラス凝集度の欠如	$\geq 90(\%)$	保守/評価/再利用/読解が難しい

目次

1. 品質監理本部紹介
2. 静的解析ツール導入の経緯と課題
3. 課題の分析と対応策
- 4. 見える化の実現**
5. 改善効果
6. 改善活動の評価と今後の課題

4.1. 見せる化：見せ方の工夫（課題・実績）

並べる

絶対値による順位を複数個、並べて示す
（メトリクスツリーマップの相対値比較を補完）
…何が悪いの？に答えるため
→しきい値を追加して表示

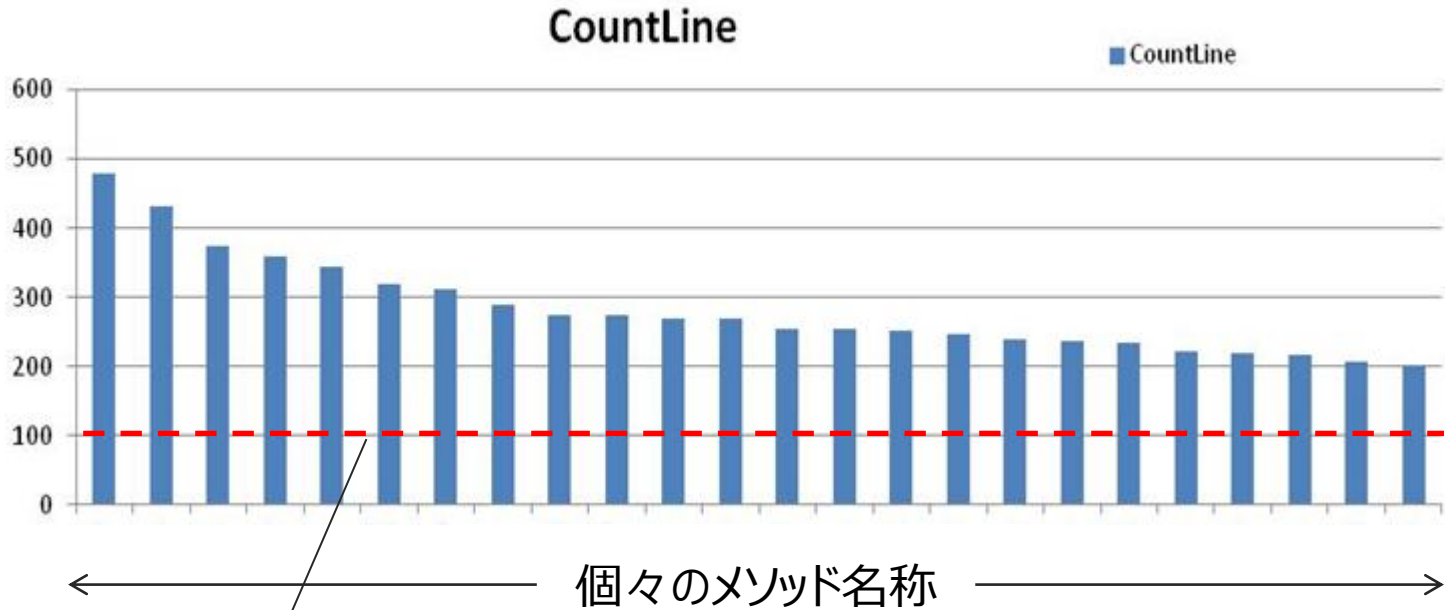
比べる

規模の大きなOSSのプログラムと比較する
リファクタリングの前後（Before/After）を比べる
…OSSと性格の違い、リファクタリング後には直せない
→過去の解析結果を整理し、比較

区別
する

文献から複雑度と規模にそれぞれしきい値を設定
双方がしきい値よりも大：危険ゾーンで示す
…規模よりも複雑度の方が、保守の障壁ではないか
→保守難度を追加

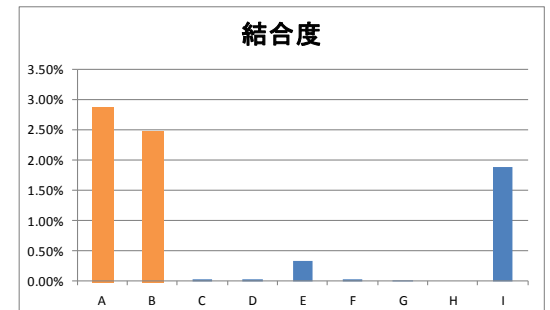
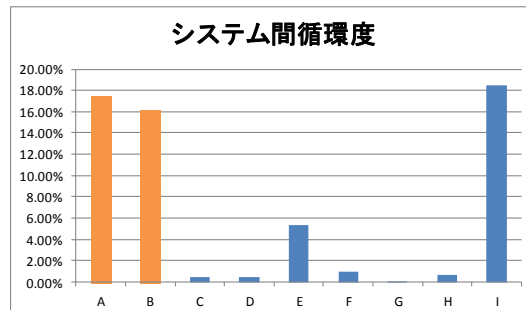
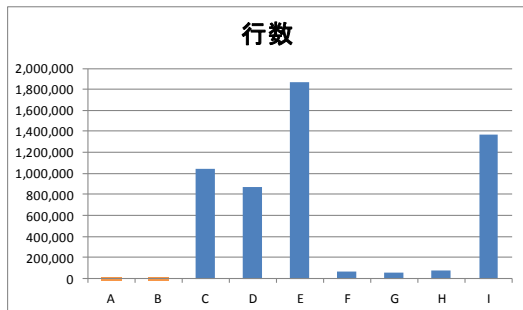
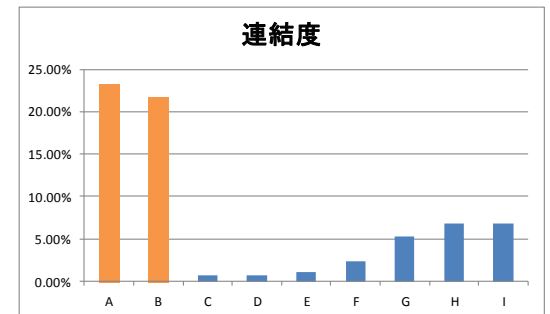
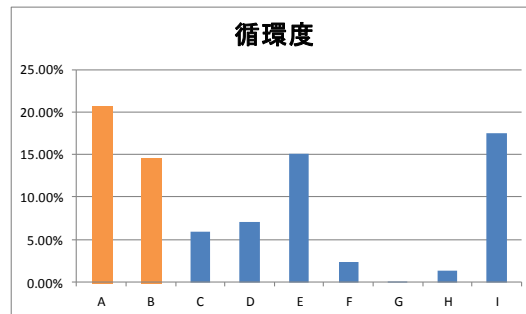
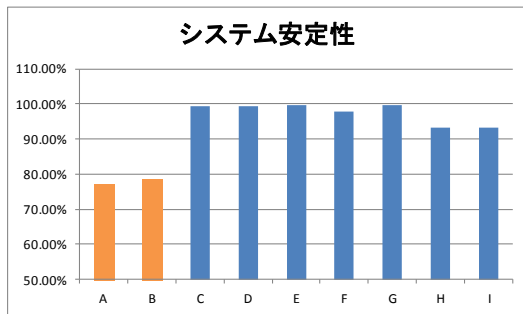
4.2. 並べる：絶対値による降順としきい値



一般的には、メソッドの行数は100以下が望ましい。
メソッドの分割を推奨。

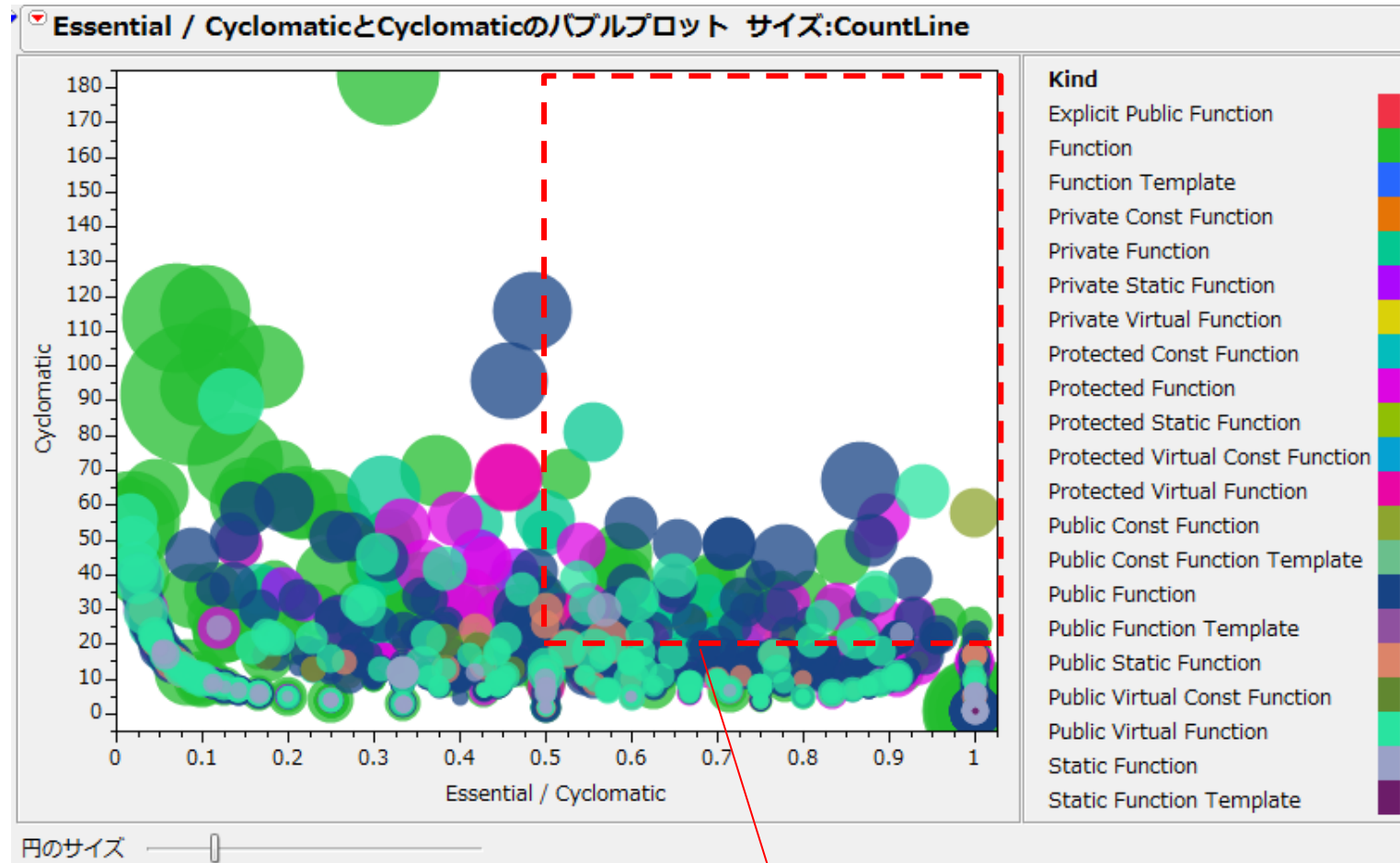
4.3. 比べる：過去の解析結果

	A	B	C	D	E	F	G	H	I
システム安定性	77.10%	78.57%	99.25%	99.39%	99.69%	97.64%	99.48%	93.17%	93.15%
影響度平均	14.43	14.57	18.87	13.14	3.95	5.05	10.23	20.98	229.54
連結度	23.27%	21.75%	0.75%	0.72%	1.06%	2.37%	5.27%	6.86%	6.85%
結合度	2.87%	2.46%	0.02%	0.03%	0.34%	0.02%	0.01%	0.00%	1.88%
循環度	20.64%	14.57%	5.95%	7.13%	15.10%	2.34%	0.10%	1.30%	17.52%
システム間循環度	17.46%	16.18%	0.48%	0.49%	5.40%	0.94%	0.10%	0.65%	18.45%
行数	14,437	15,021	1,044,019	874,357	1,866,931	65,843	52,169	73,258	1,371,047



4.4. 区分けする：

複雑度と保守難度の危険ゾーン



赤枠部分が、保守が特に難しい
(誤った修正を行う可能性が高い)

目次

1. 品質監理本部紹介
2. 静的解析ツール導入の経緯と課題
3. 課題の分析と対応策
4. 見える化の実現
- 5. 改善効果**
6. 改善活動の評価と今後の課題

5.1. その後の現場の意見

どこが複雑で規模が大きいかをわかっているつもりでいたが、具体的な数値では掴めていなかった。

また、しきい値を超えたものがどれだけあるのかは、わからなかった。

他の解析結果と比較できると、自分のプログラムがどの程度の作りなのか、考えさせられる。良いか悪いかではなく、値が突出しているものは気になる。

危険ゾーンはわかりやすい。またそのゾーンにどの程度の規模のものがいくつあるのか、明確である。このゾーンに入らないようにしたい。



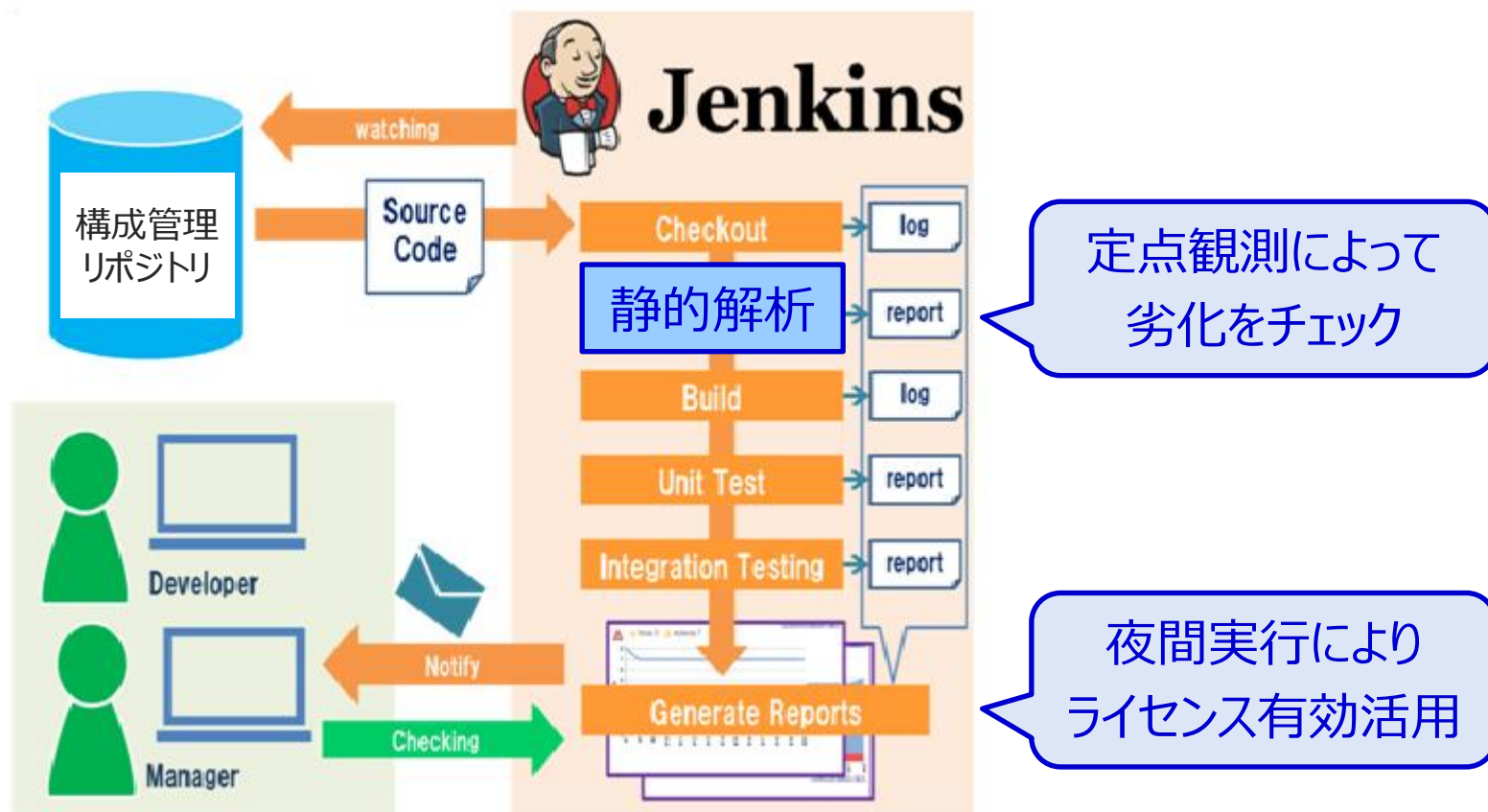
現場の気づきが現場の改善を促した



5.2. 現場の改善1：

継続的統合環境への組み入れ

- 継続的な統合（CI）環境でメトリクス収集、確認



5.3. 現場の改善2 :

ルール化による劣化の防止

■ Understandでの関数・クラスの指標値例

指標名	上限値	製品共通化	補足
CyclomaticModified	20	対象	サイクロマチック複雑度
MaxNesting	4	対象	ネストの最大数
CountLineCodeExe	100	対象	実行可能コード行数
[Essential]	10	任意	本質的複雑度
[CountClassCoupled]	30	任意	クラス結合度

■ 開発時のルール

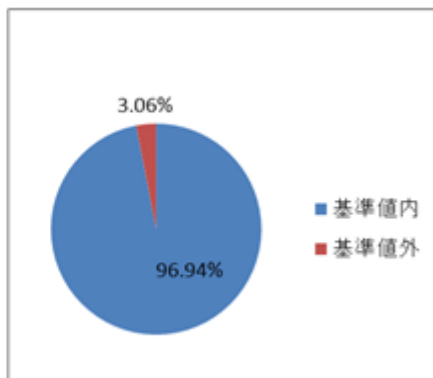
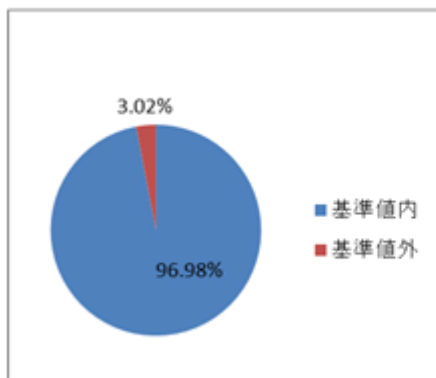
- 関数の新規作成時は指標の上限値を超えないコードを作成する
- すでにこの値を超えている関数の修正時は値を減らす(努力目標)
- すでにこの値を超えている関数の修正時は値を悪化させない(義務→努力目標)

5.4. 現場の改善3 :

バージョン間による劣化の確認

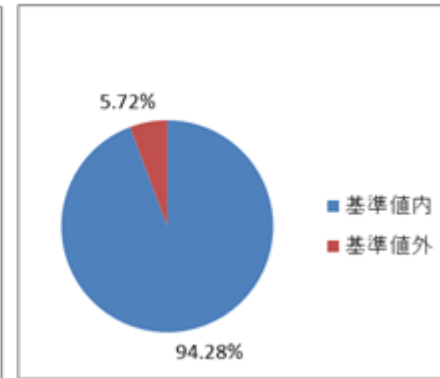
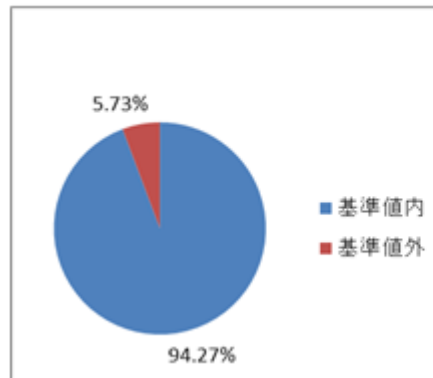
Cyclomatic基準値超過割合

	前バージョン	新バージョン
基準値内	20,746	20,941
基準値外	645	660



MaxNesting基準値超過割合

	前バージョン	新バージョン
基準値内	20,773	20,980
基準値外	1,263	1,274



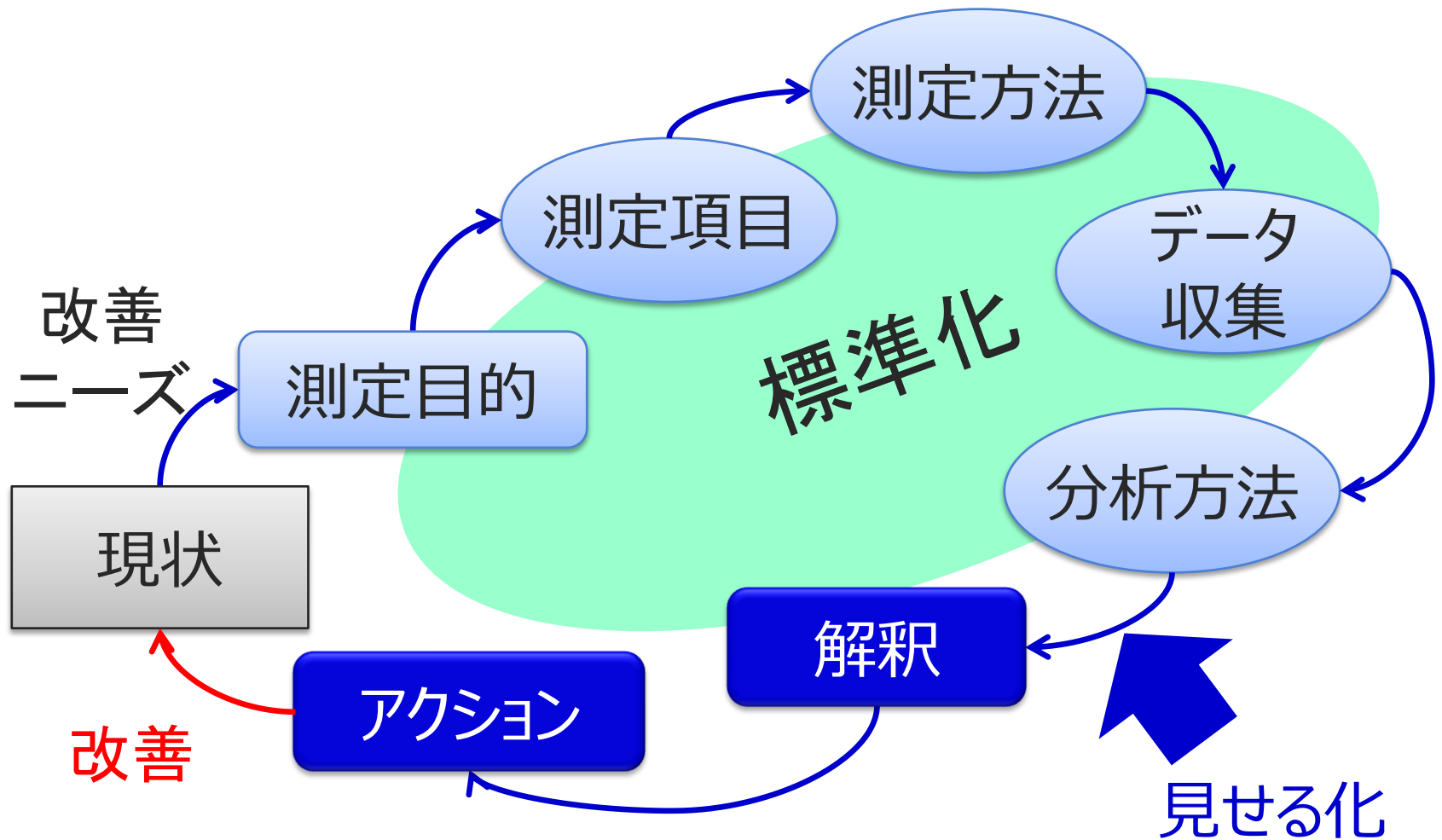
<結果考察>

- ①PG工程終了付近でメトリクス評価を始めたこともあり、リファクタリング作業が出来なかった
⇒PG工程の当初から監視する必要あり、また開発者の意識付けが重要
- ②全体としてメトリクス値は大きな変化なし
⇒販売実績の長い製品は簡単に修正できない事情もあり、基準外を削減する事は難しいと考える。逆に現状を維持する事に注力する(基準値を高くするなどの検討も必要かも)
- ③境界線付近に一番多く分布
⇒今後のエンハンス時に改修対象であれば、意識した改善を行っていく

目次

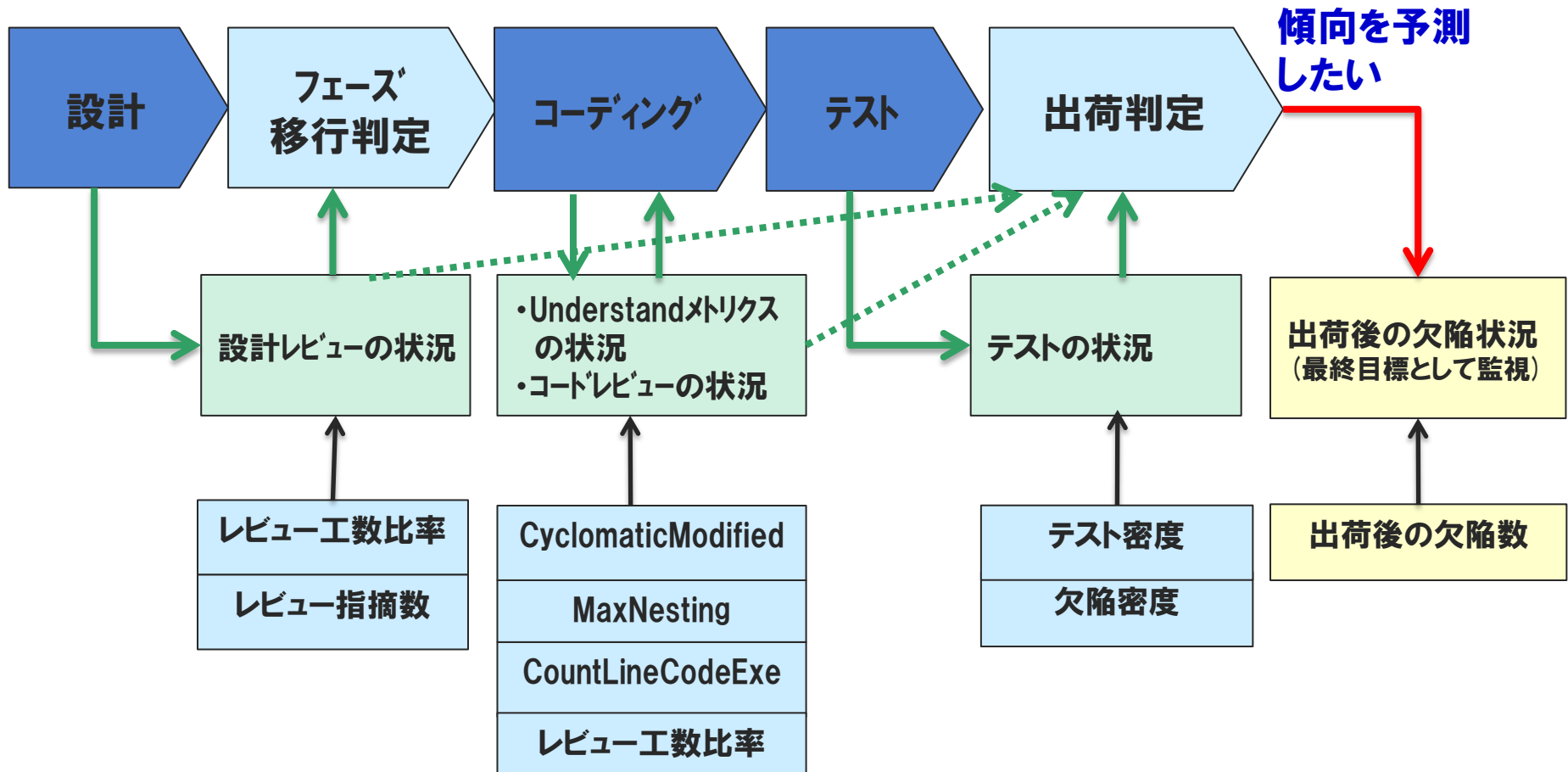
1. 品質監理本部紹介
2. 静的解析ツール導入の経緯と課題
3. 課題の分析と対応策
4. 見せる化の実現
5. 改善効果
- 6. 改善活動の評価と今後の課題**

6.1. 見せる化→気づき→アクションへ

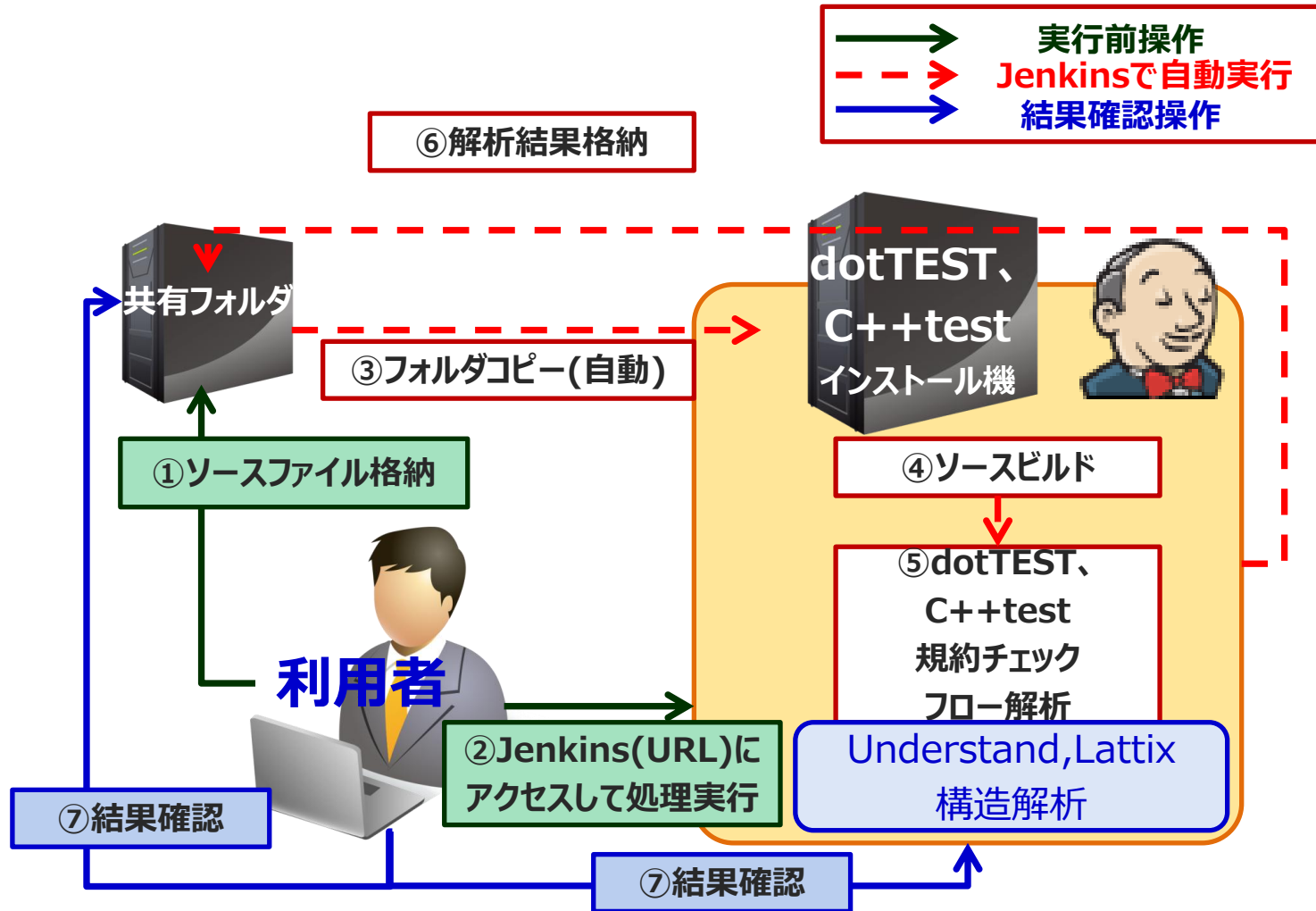


6.2. ルール化→プロセスへの組み入れ

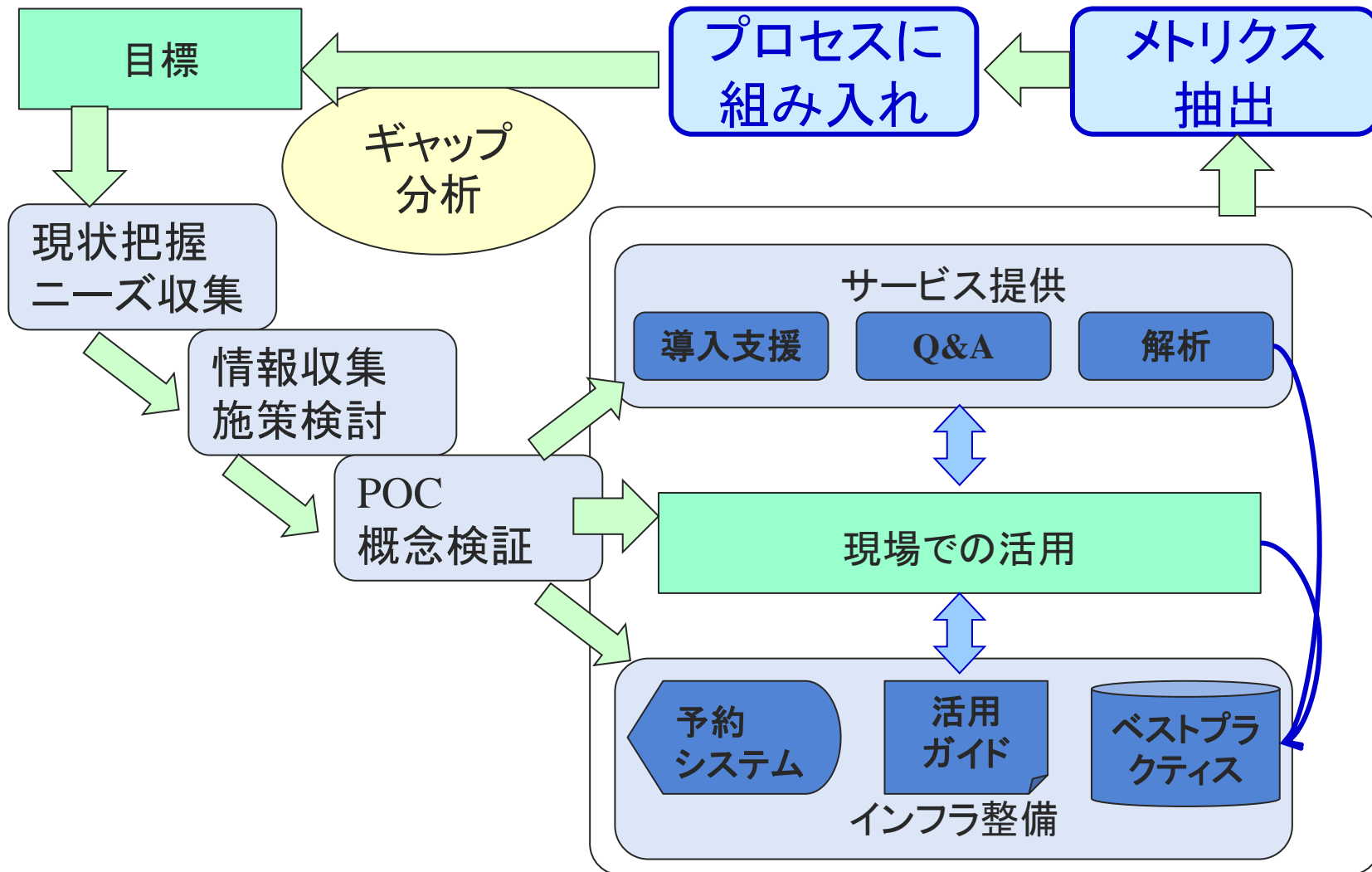
- 移行判定の条件への組み入れ、データ蓄積による予測へ



6.3. 組織的なサポートとデータ収集



6.4. 継続的な改善へ



Q&A



静的解析結果の見せる化

～並べて、比べて、区分けして、気づきを促す～

Thank you

Canon