最近の静的解析の立ち位置

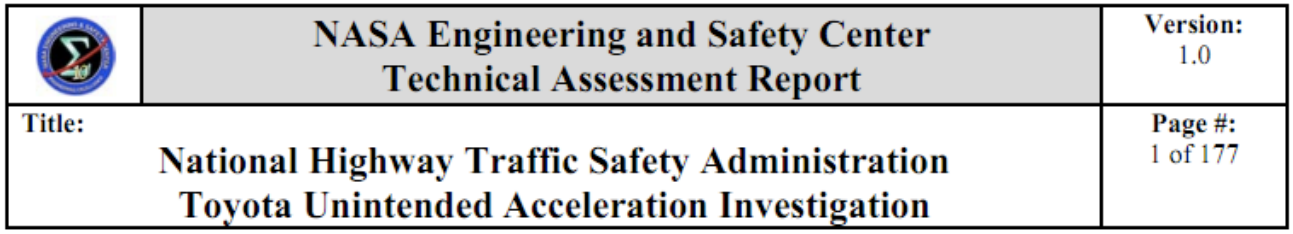
コベリティ日本支社

coverity®

静的解析が注目される背景

NASAによる日本車製乗用車の急加速問題調査

- 2011年2月に調査結果を公開
  - http://www.nhtsa.gov/staticfiles/nvs/pdf/NASA-UA_report.pdf
  - 電子制御システムの問題ではないことがわかり、疑いが晴れた



	NASA Engineering and Safety Center Technical Assessment Report	Version: 1.0
Title: National Highway Traffic Safety Administration Toyota Unintended Acceleration Investigation		Page #: 1 of 177

- ETC (Electric Throttle Control) のシステム設計・実装に意図していない不具合が含まれているかを調査

実装ソースコードの解析に
Coverity 静的解析を利用

coverity®

静的解析が注目される背景

- ソフトウェアに起因する産業機器の不具合は少なくない

米国FDA（食品医薬局）は、
2010 年前半に、

不具合による 23 件の
デバイスのリコールを行った

これらのリコールのうち、
少なくとも 6 件は、
ソフトウェアの不具合だった。

coverity®

静的解析が注目される背景

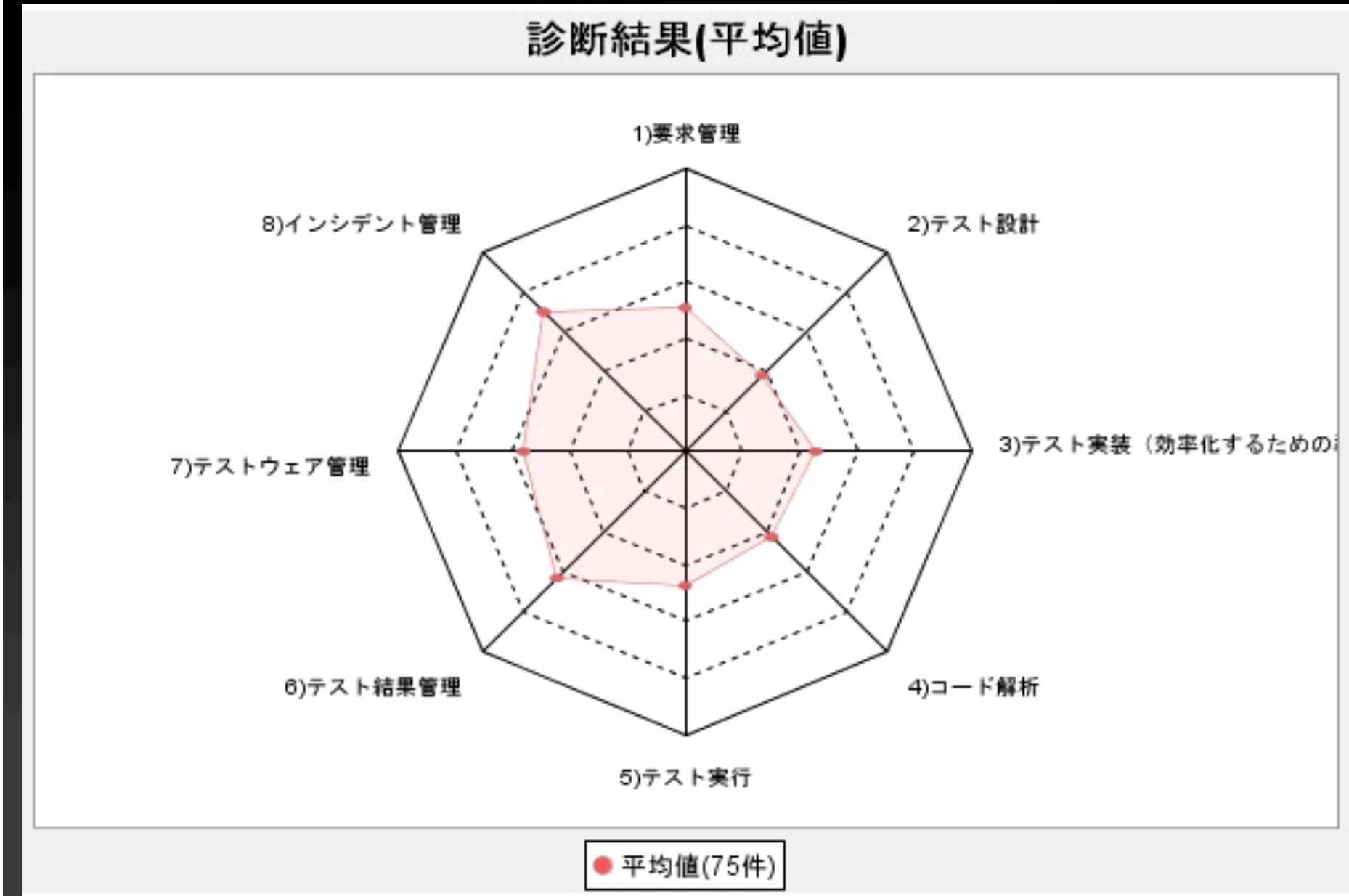
- DISA STIG：Defense Information Systems Agency Security Technical Implementation Guide は防衛関連システムのセキュリティガイドとして利用

- DO-178B：Federal Aviation Administration (FAA) は、航空機のソフトウェアに対し DO-178B をコンプライアンスとして利用

- FDA Guidance: General Principal of Software Validation は、US 医療機器ソフトウェアに関するコンプライアンス

- ISO26262: 自動車業界向けのソフトウェア開発に対するコンプライアンス

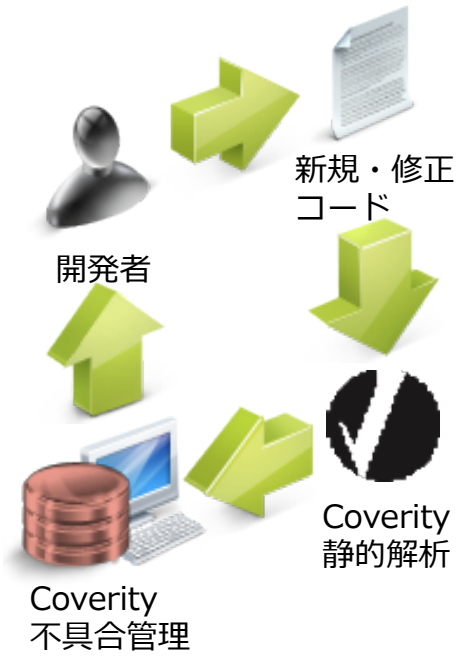
でも....


診断結果(平均値)
1)要求管理
2)テスト設計
3)テスト実装（効率化するための
4)コード解析
5)テスト実行
6)テスト結果管理
7)テストウェア管理
8)インシデント管理
平均値(75件)

ツールWGの診断の結果。

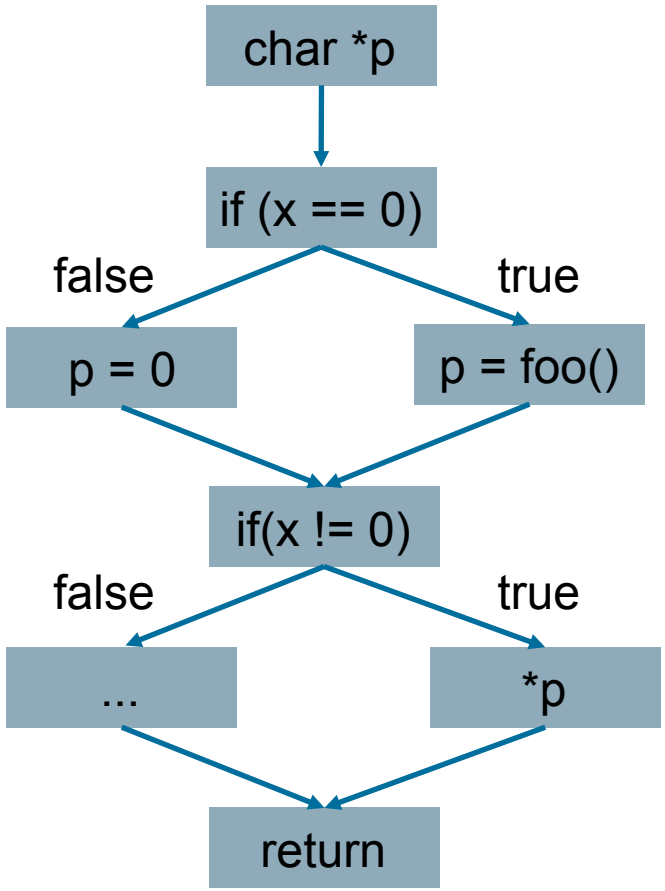
コード解析の
利用自体が低そう？

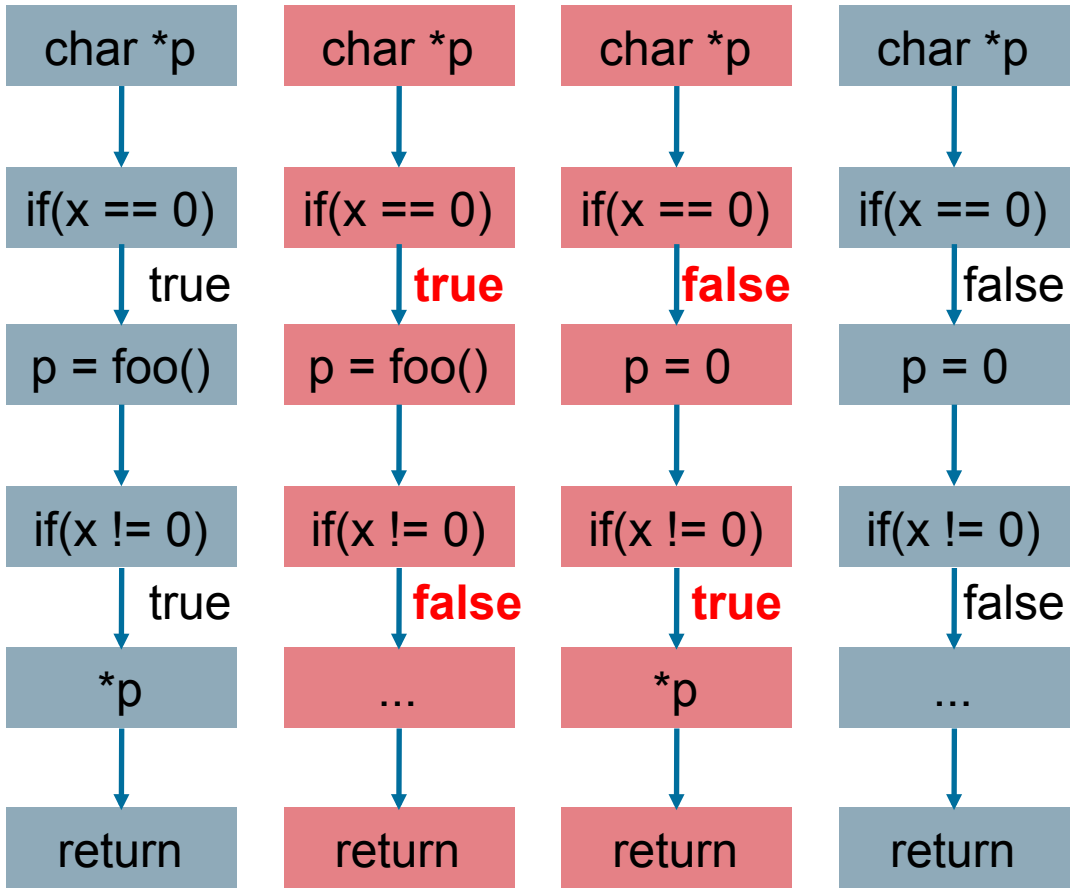
というわけで
おさらいから



coverity®

コードから検出される不具合って





coverity®

検出される欠陥の種類（主なもの）

C/C++

並列処理の問題
- デッドロック
- 競合状態
- ブロック呼び出しの誤用

パフォーマンスの低下
- メモリ・リーク
- スタックの使用度合い

クラッシュの原因
- Null ポインタの間接参照
- ポインタの解放後のメモリ使用
- 二重解放
- 配列の新規作成と削除の不一致

セキュリティ上の脆弱性
- バッファ・オーバーフロー
- API エラー処理

API などの不適切な使用
- STL の誤使用
- API エラー処理

プログラムの不正な動作
- デッドコード
- 未初期化変数
- 負の変数の無効な使用
- MISRA-C キャスト違反

Java/C#

並列処理の問題
- デッドロック
- 競合状態
- ブロック呼び出しの誤用

パフォーマンス低下/スケーリングの制限
- リソース・リーク
- データベース接続リーク

クラッシュの原因
- null の間接参照
- リソースの解放後の操作

プログラムの不正な動作
- デッドコード
- 未初期化変数
- 負の変数の無効な使用

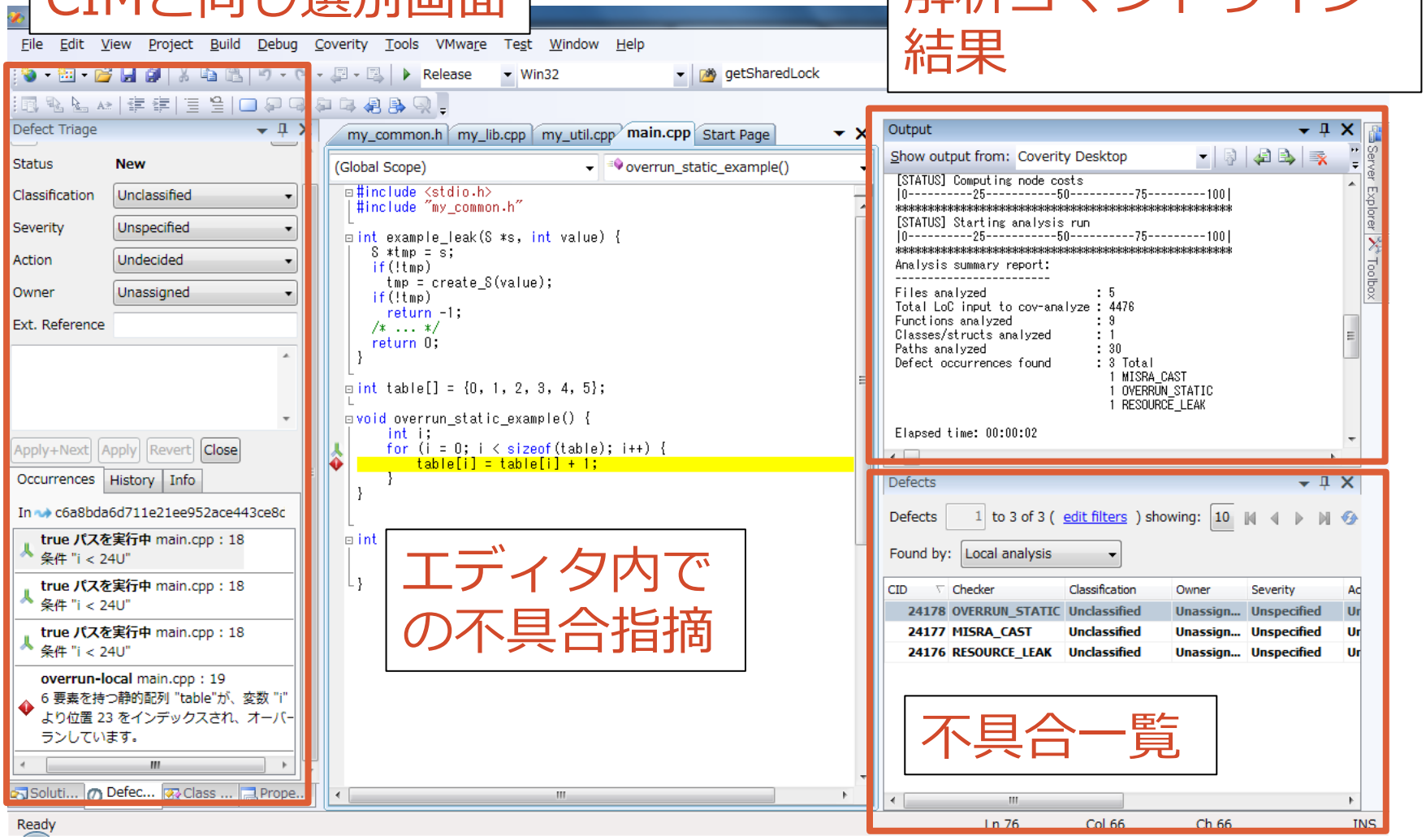
API の不適切な使用
- API エラー処理

FindBugs で検知された不具合

Visual Studioの画面例

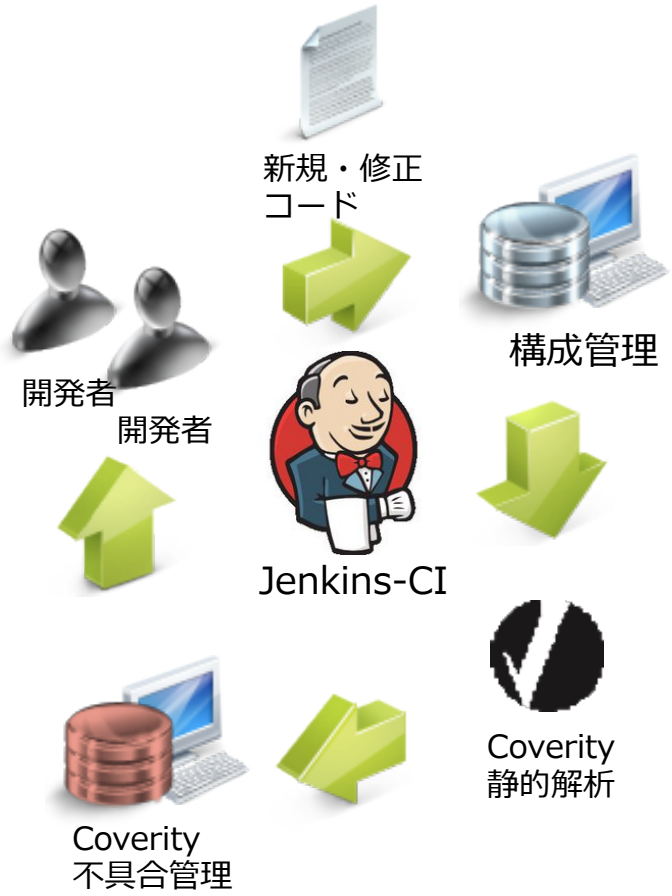
CIMと同じ選別画面

解析コマンドライン結果



coverity

チーム全体の
最新のコード品質を
得るには？


新規・修正
コード
構成管理
開発者
開発者
Jenkins-CI
Coverity
静的解析
Coverity
不具合管理

coverity®

継続的な統合 + 静的解析

開発者が
Check in

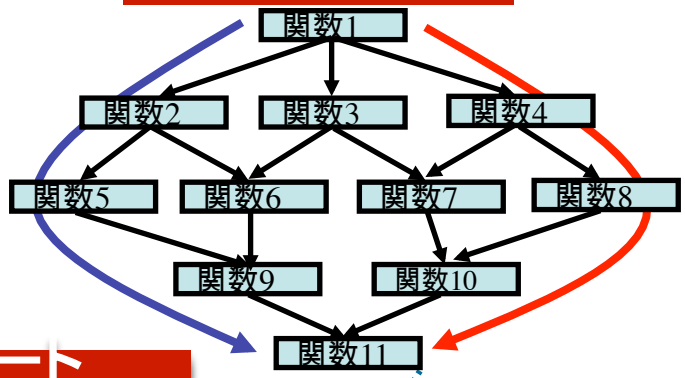
構成管理

Jenkins-CIが
Check out

開発者

```
562    if (len == 0)
562    if (len == 0)
563        buf = NULL;
564    else
565    {
566        buf = malloc(len);
567
568        if (buf == NULL) {
569            read_error = TRUE;
570            return (READSTAT);
571        }
572    }
575    /* productA V1.0 */
576    rs_read_chars(inf, buf, len);
577
```

静的解析



不具合管理とレポート
(CIM)

不具合箇所の
抽出

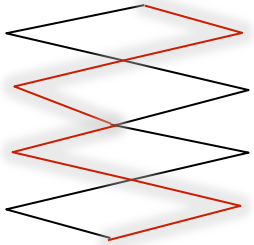
```
long int SomeFunction();
/* int OtherFunction(); */

/* int */ CallingFunction()
{
    long int test1;
    register /* int */ test2;

    test1 = SomeFunction();
    if (test1 > 0)
        test2 = 0;
    else
        test2 = OtherFunction();

    return test2;
}
```

不具合箇所に至
るパスの抽出



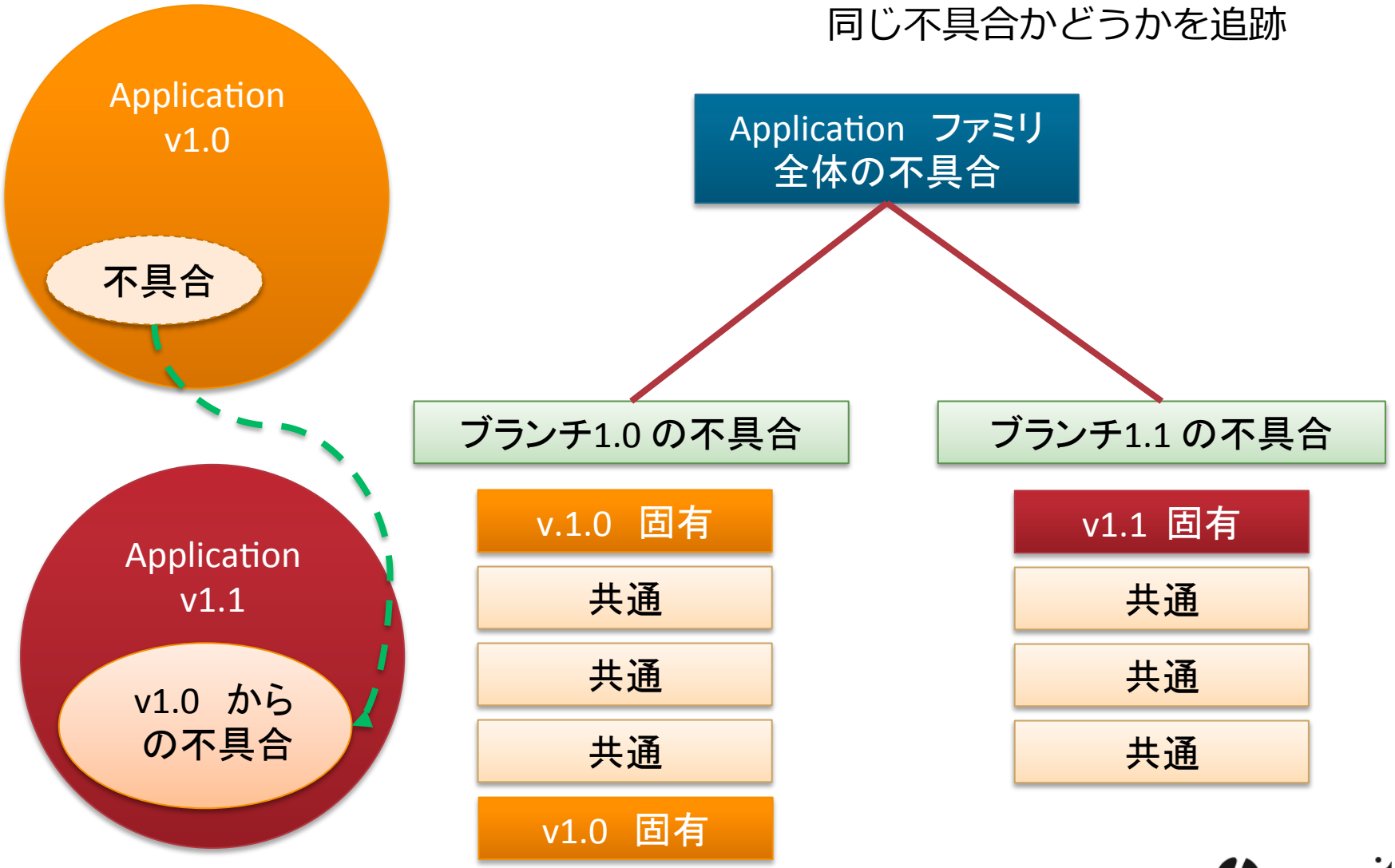
Jenkins-CI
または
CIM
よりメール通知

Jenkins-CIが
解析を実行

coverity

派生開発、第三者コードを考慮したインパクト分析

不具合の発生する制御パスを比較し
同じ不具合かどうかを追跡

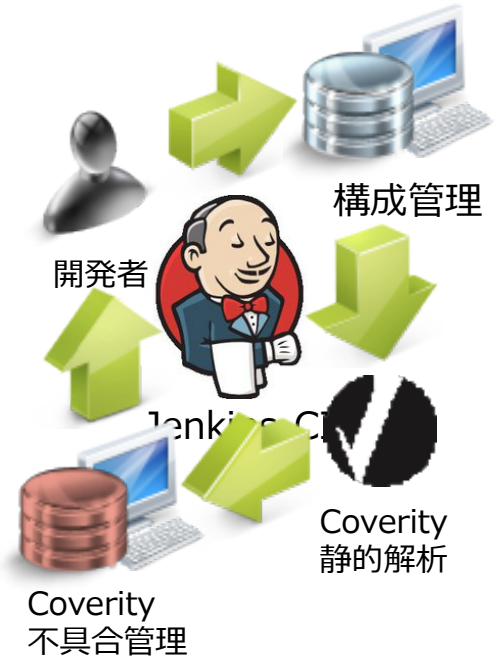


coverity®

継続的にコード品質
を確認できる～

このデータを
を組織運営に活かす

開発組織A



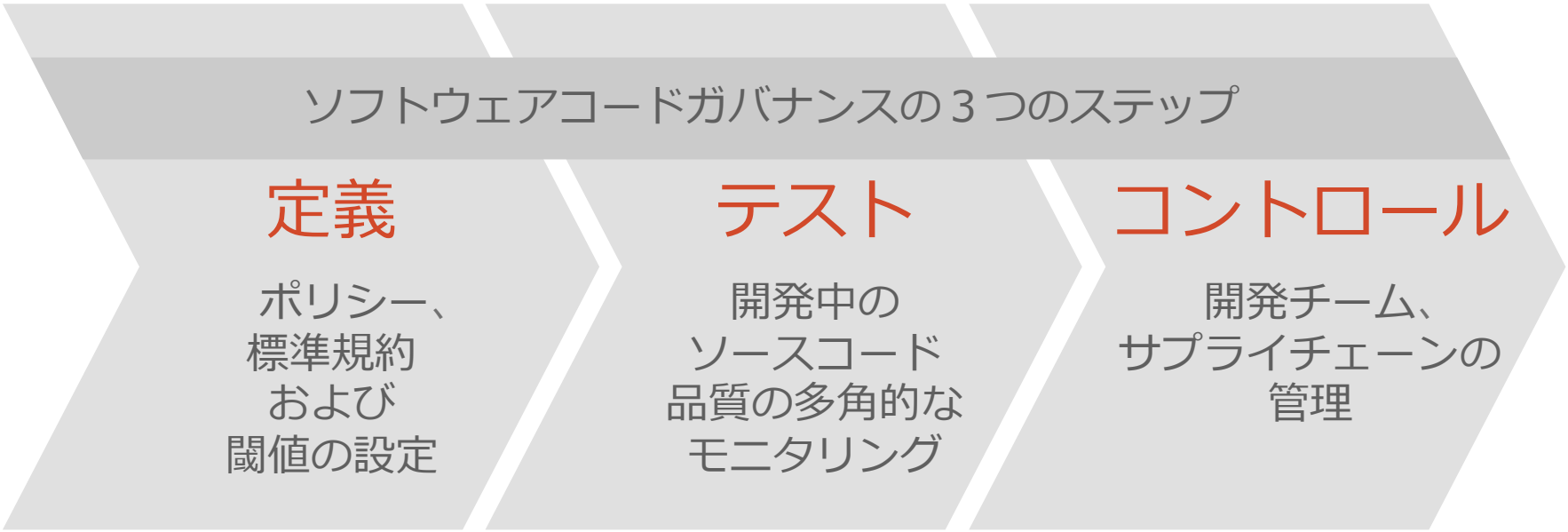
開発組織B

協力会社X

データウェア
ハウス

coverity®

開発状況の可視化と
ソフトウェアコードのガバナンス



CIM（複数も可）からの情報の収集とモニタリングを行い
企業レベルでのソフトウェアコードのガバナンスを実現

coverity®

ソフトウェアコードガバナンス

シナリオ１：プロジェクトレベルのリスクマネージメント

プロジェクト内の各コンポーネントがどのような状態にあるかを随時監視

シナリオ２：サプライチェーンリスクマネージメント

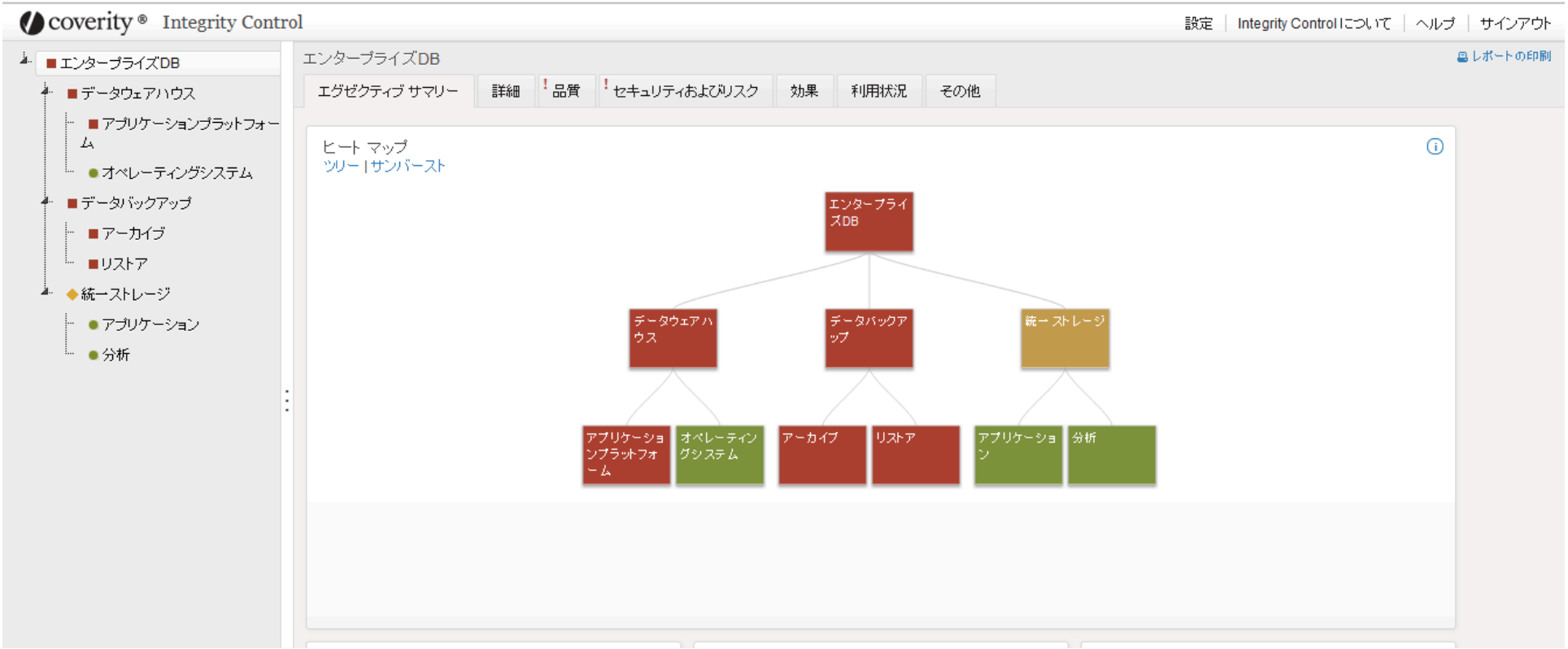
外部組織からの成果物に対する、品質チェック、セキュリティ規準の厳守チェックなど

複数の組織を階層的に管理可能

シナリオ３：中央管理体制の確立

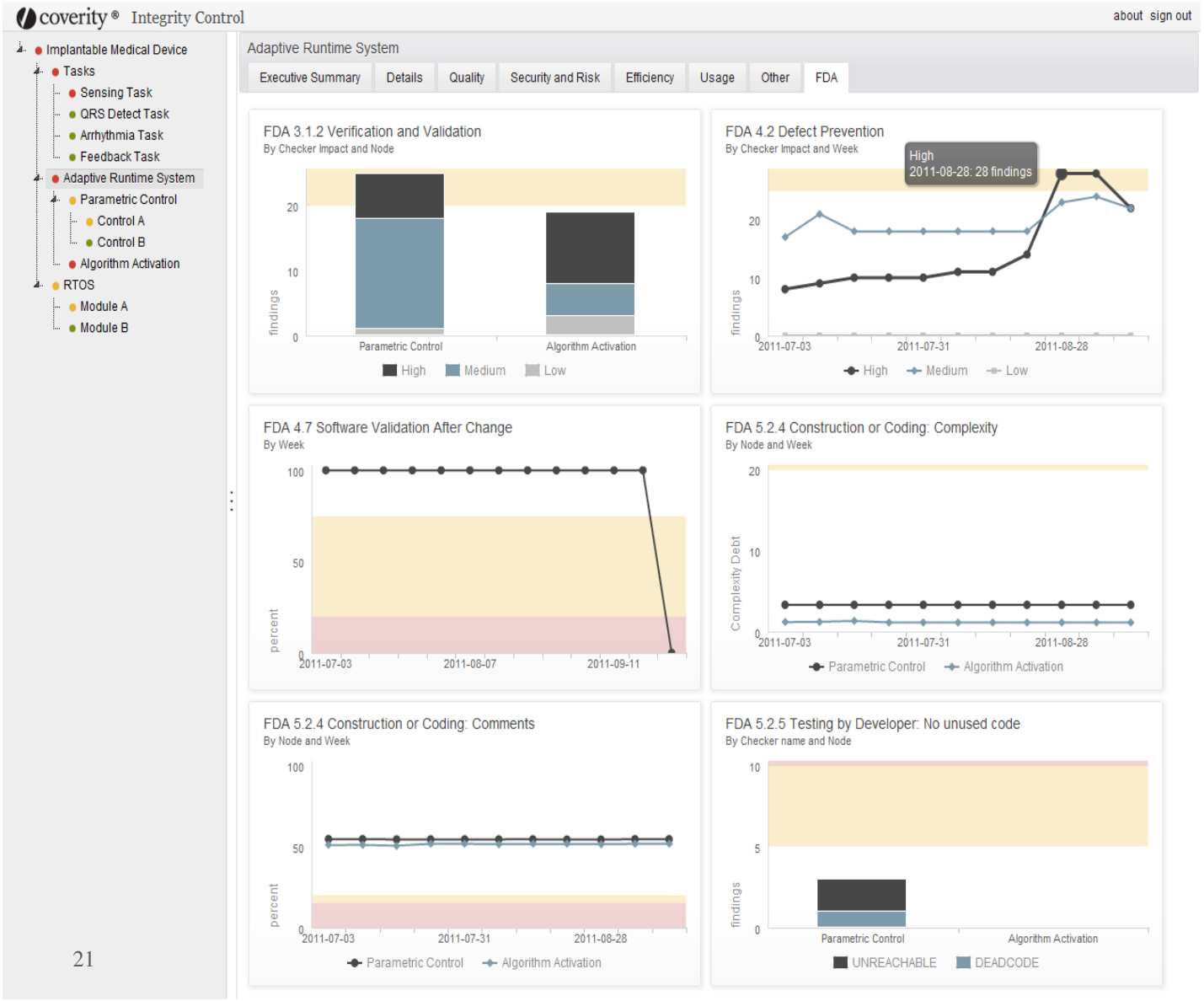
全開発組織の組織単位でのソフトウェアコード品質の状況を、VPレベルの視点からレポート

coverity®



coverity®

FDA用テンプレート



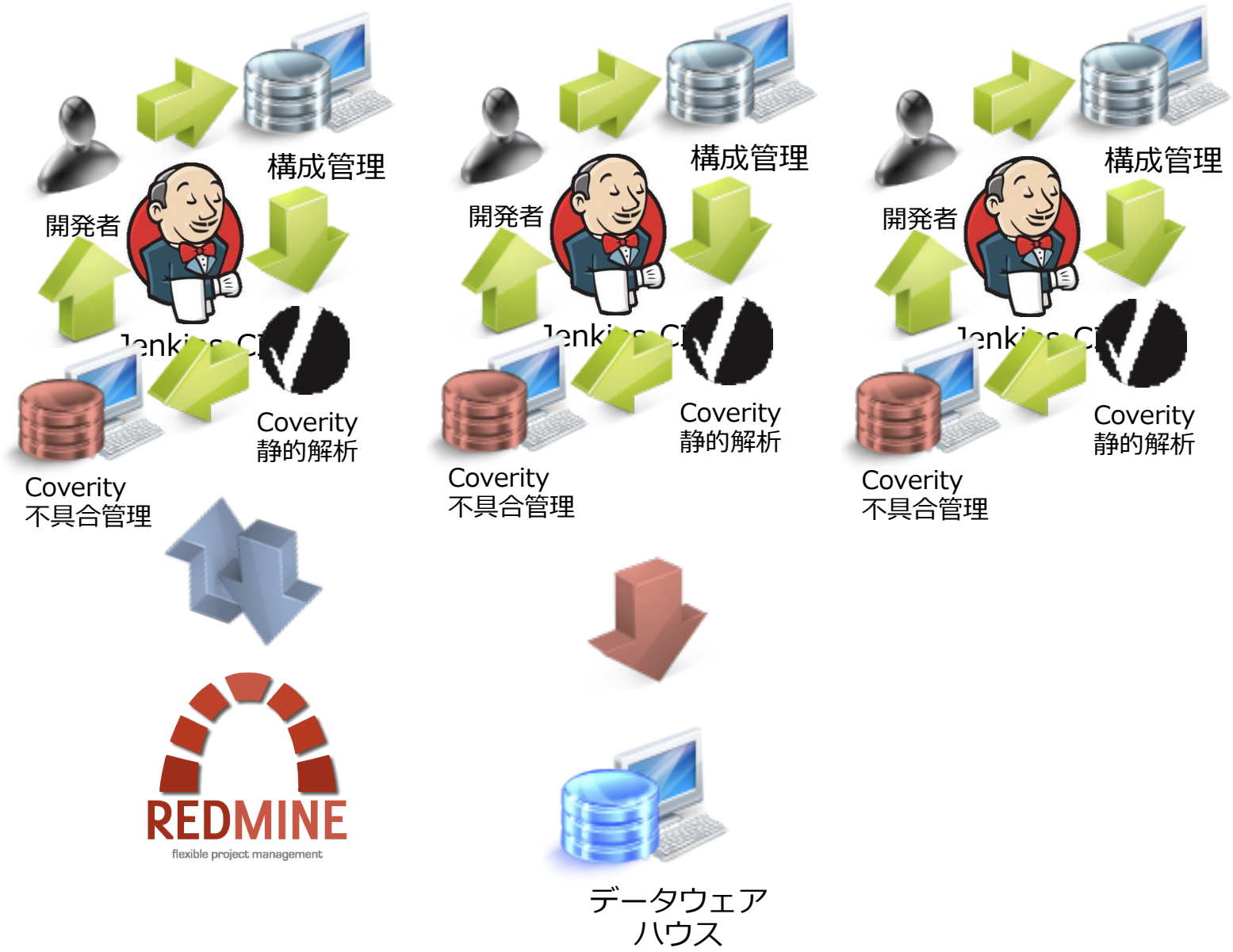
医療機器プロジェクトであれば、FDA ガイドラインに則したポリシーテンプレートを利用できる

ポリシー設定により、コンプライアンスに関するメトリクスのトラッキングを容易にする

coverity®

コードレベルなら
モニタリングできるけど

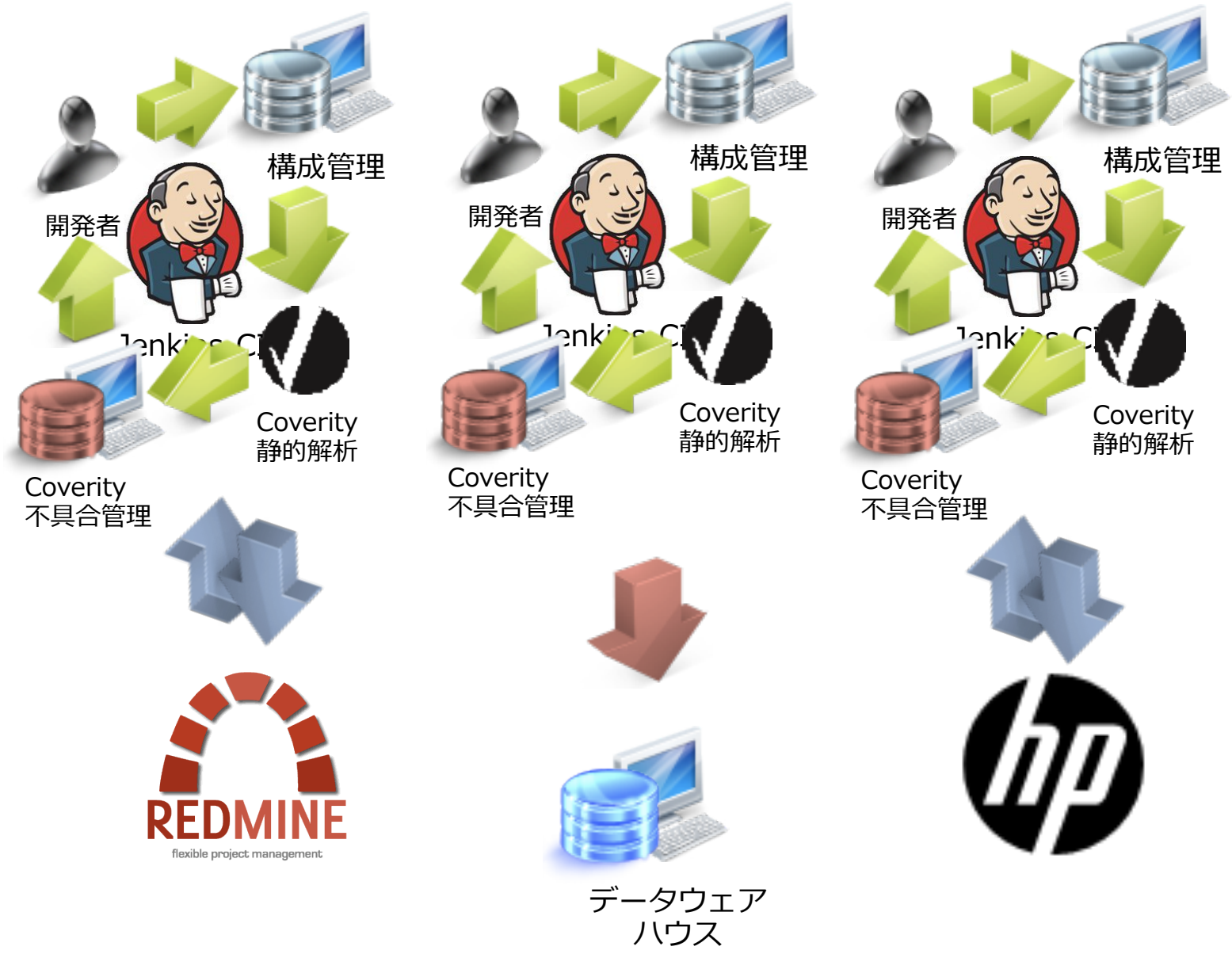
他の不具合も含めた
管理は？


構成管理
開発者
Coverity
静的解析
Coverity
不具合管理
構成管理
開発者
Coverity
静的解析
Coverity
不具合管理
構成管理
開発者
Coverity
静的解析
Coverity
不具合管理
REDMINE
flexible project management
データウェア
ハウス

coverity®

不具合だけでは足りない……

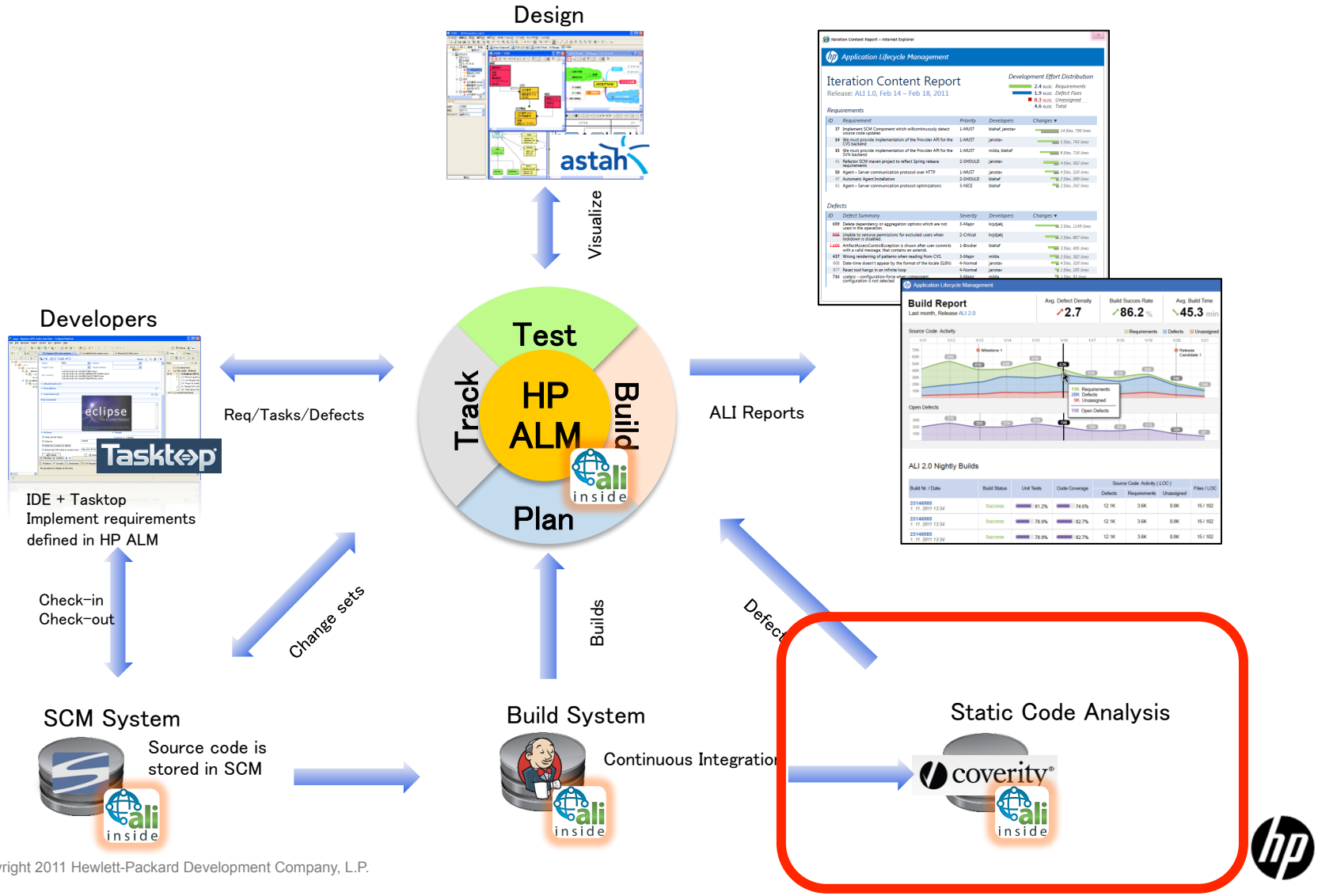
真の変更管理は？
ALMとの連携


構成管理
開発者
Coverity
静的解析
Coverity
不具合管理
REDMINE
flexible project management
構成管理
開発者
Coverity
静的解析
Coverity
不具合管理
データウェア
ハウス
構成管理
開発者
Coverity
静的解析
Coverity
不具合管理

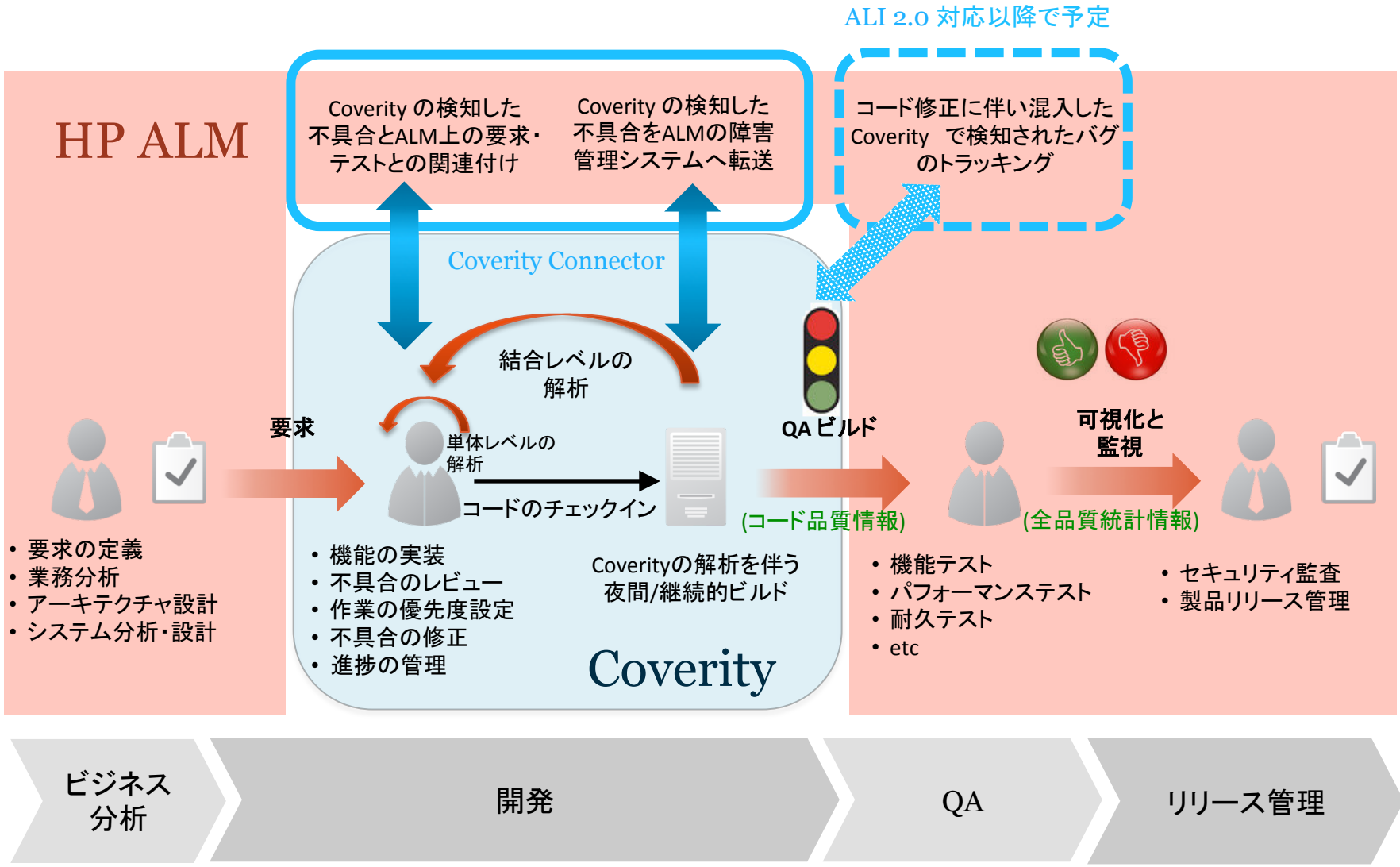
coverity®

静的解析ツール連携

Coverity社　Static Analysis

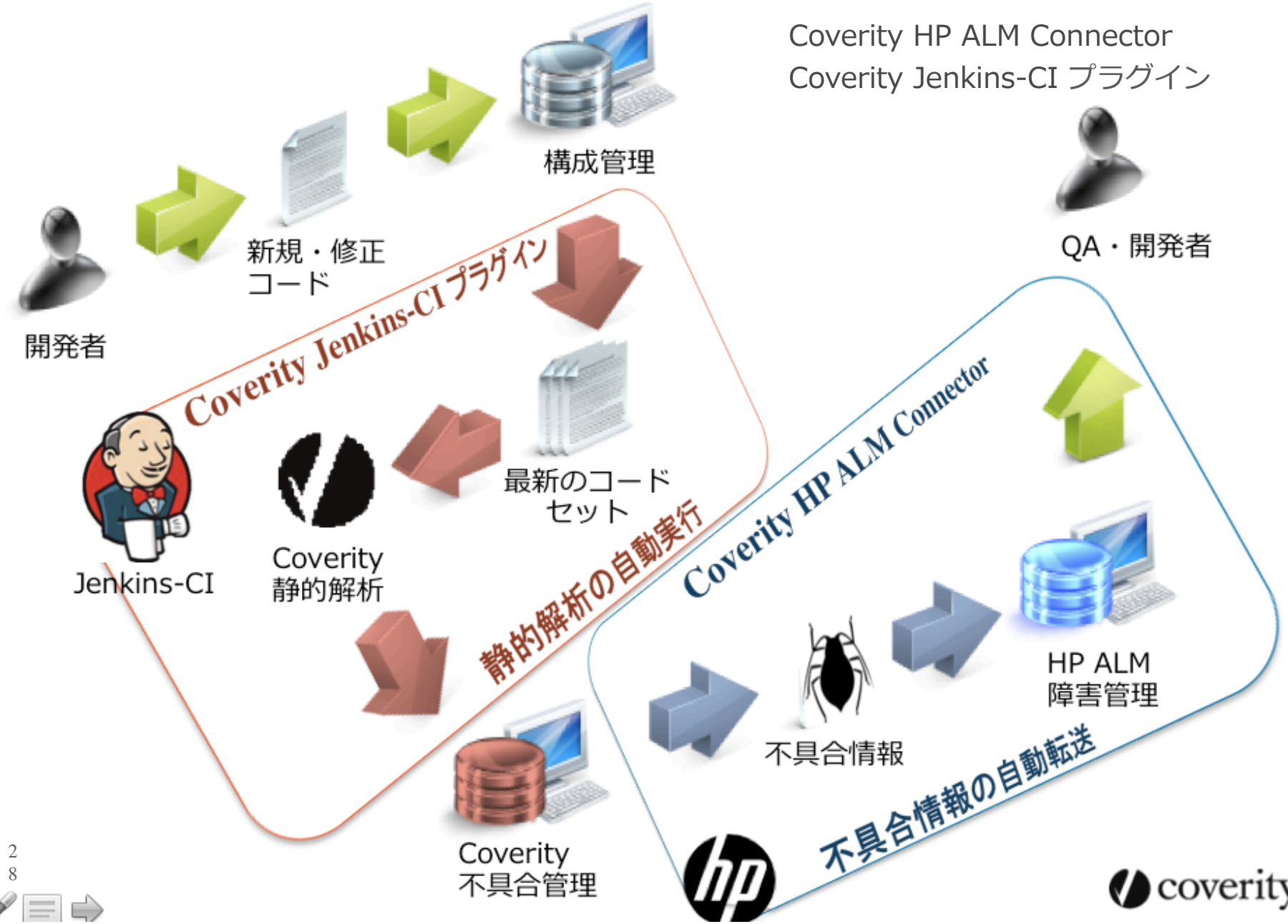


Coverity HP ALM Connector

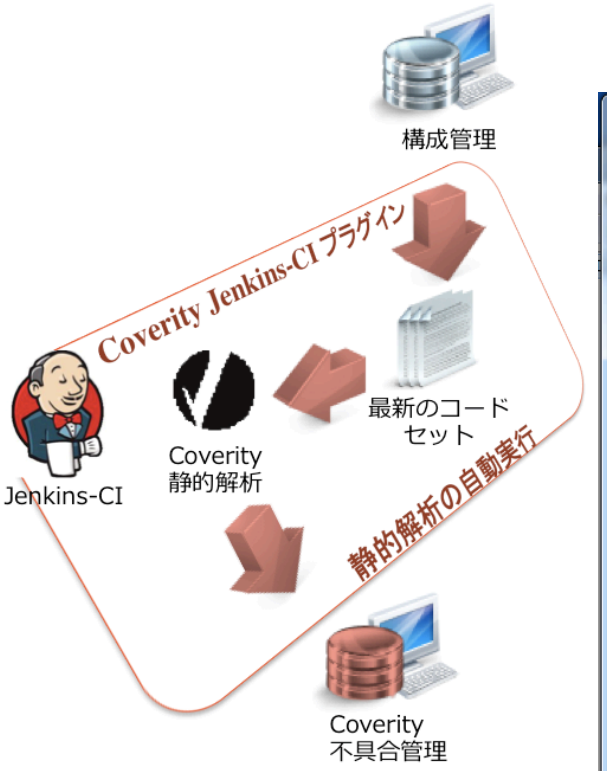


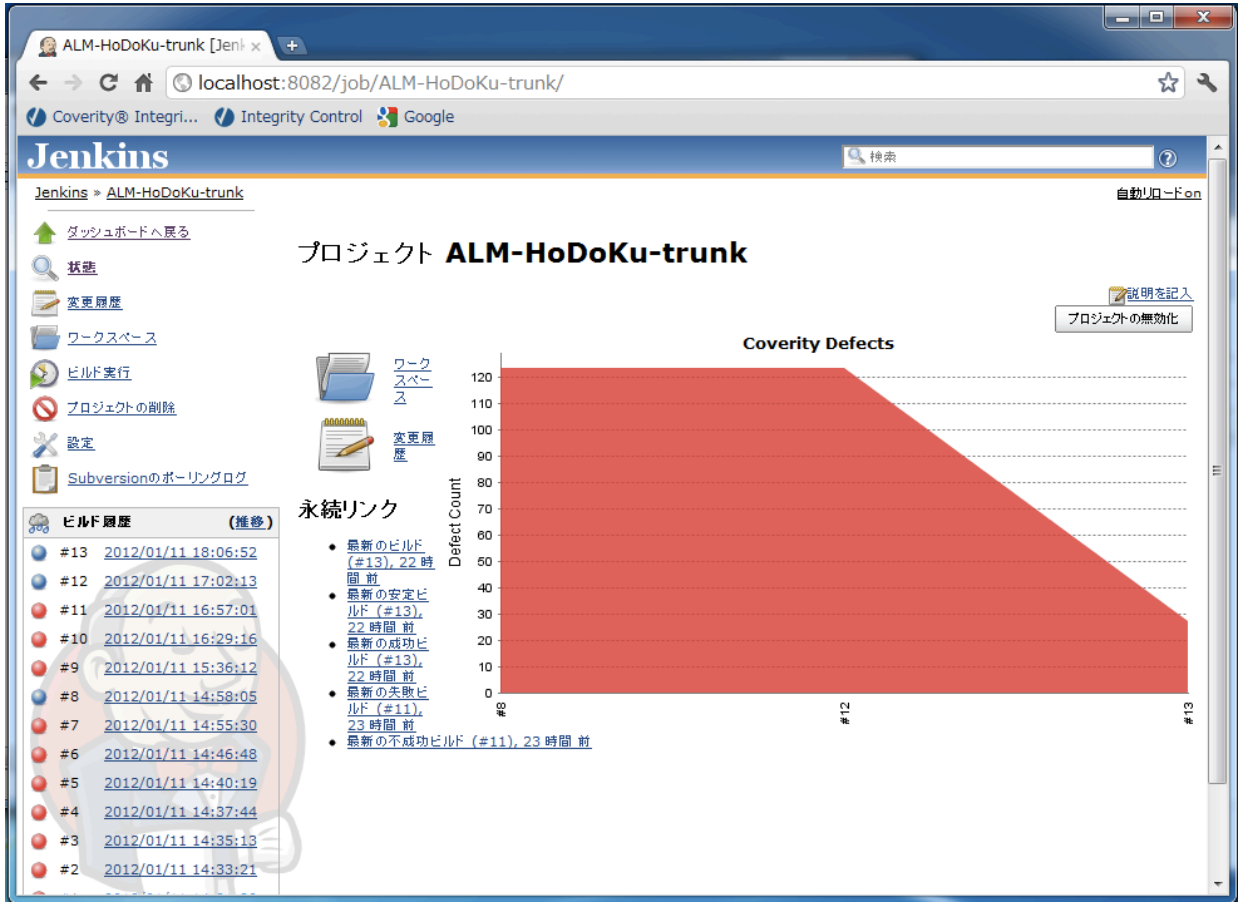
coverity®

具体的な連携フロー（不具合情報の例）



Jenkins-CI Coverity プラグイン

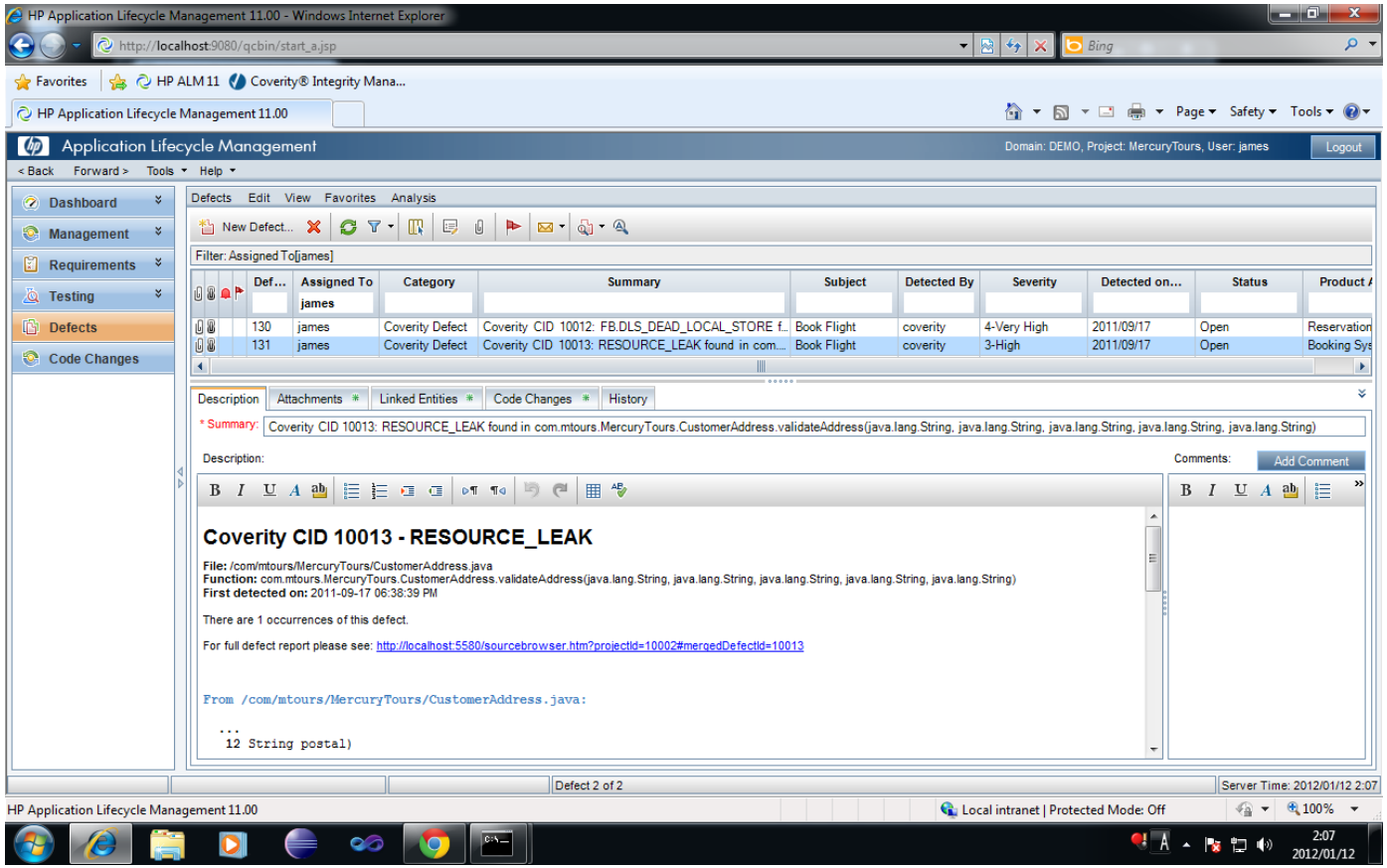




構成管理リポジトリをポーリングし、
自動的にビルド／静的解析を実行

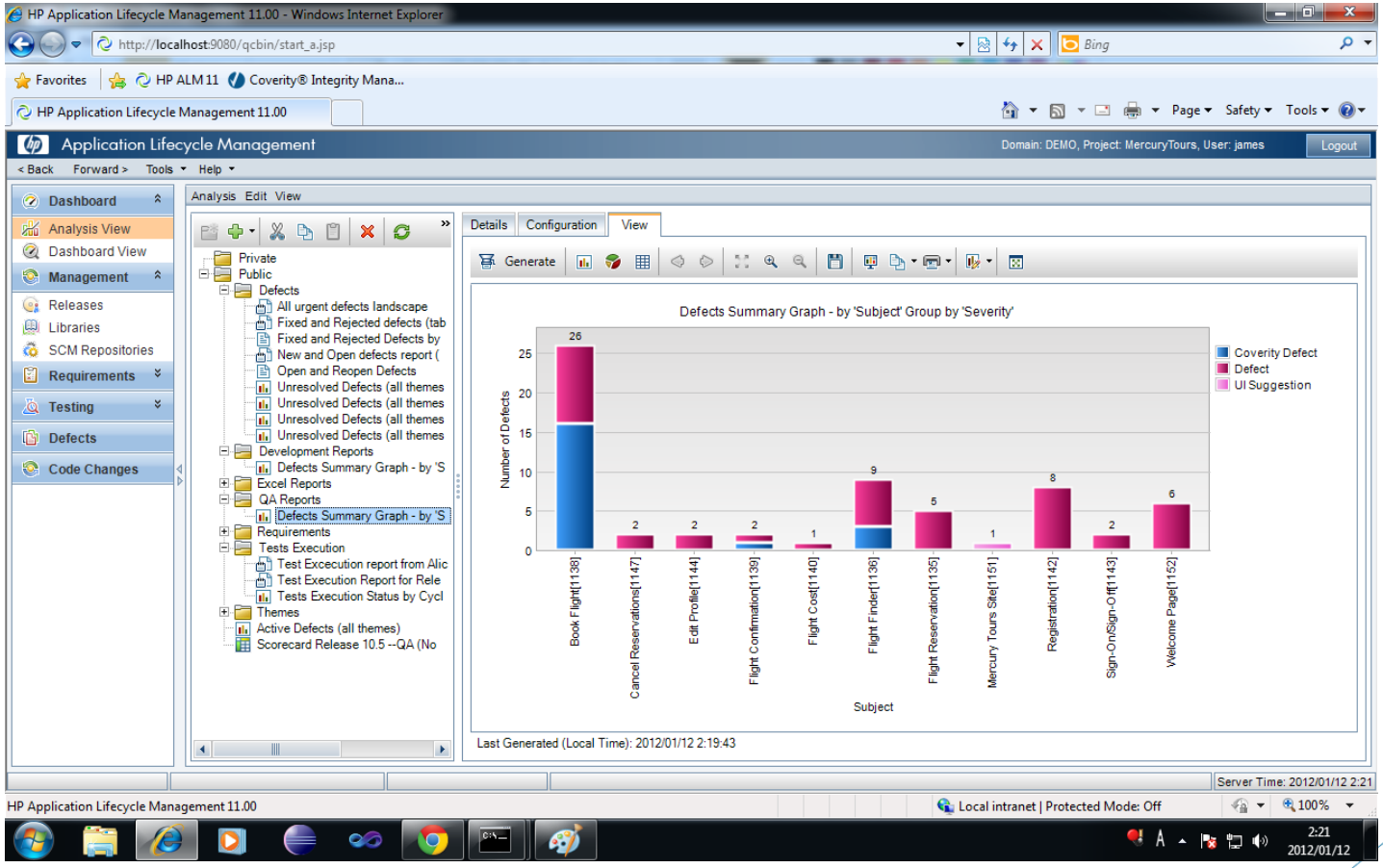
coverity®

Coverity HP ALM Connector
不具合情報の共有



不具合の詳細情報とともに、
HP ALM上に不具合除法を転送

Coverity HP ALM Connector
コード品質に関するKPIをALMで一括管理



コード品質に関わるKPIは
静的解析ツールから自動収集し
モニタリング可能

今年はどんな立ち位置？

無料お試しできます！

展示ブースにお立ちよりください