JaSST'2006

組込みソフトウェア　信頼性向上ソリューションご紹介
（コードレビュー、動的テスト、静的検証支援ツールご紹介）

株式会社エーアイコーポレーション

ツールの必要性

ソフトウェア規模・複雑さの増大

短納期化による開発期間短縮

十分なテストによる信頼性確保？

人手だけではなくツールによる効率化

信頼性向上支援ツール

- 実行時エラー自動検出ツール
  - PolySpace
- 動的テスト支援ツール
  - Cantata++
- ソースコードレビュー支援ツール
  - DAC（Development Assistant for C)

ツールと不具合の関係

- ソフトウェア不具合
  - 仕様の不具合
  - 設計の不具合
  - 実装時のミスによる不具合
    - 実行時エラー（ランタイムエラー）
      - ✓零除算
      - ✓不正なポインタ参照
      - ✓配列の境界外へのアクセス
      - ✓・・・

      → ***PolySpaceで検出***
    - 仕様とは違った動作
      - ✓定数の設定間違い
      - ✓エラー処理を忘れる

      → ***Cantata++で確認***

PolySpace 解析結果表示

example.c

```
62      static void Pointer_Arithmetic ()
63      {
64        int tab[100];
65        int i, *p = tab;
66
67        for(i = 0; i < 100; i++, p++)
68          *p = 0;
69
70        if(random_int() == 0)
71          *p = 5; /* Out of bounds */
72
73        i = random_int();
74        if (random_int())  *(p-i) = 10;
75
76        if (0<i && i<=100)
77        { p = p - i;
78          *p = 5;      /* safe pointer access */
79        }
80      }
```

PolySpace

静的に解析
プログラムの実行無し

緑：エラーが起こらない

赤：必ずエラーとなる

オレンジ：場合によってエラー

信頼性の向上

テストケースによるテストの結果、エラーが無いと確認される組み合わせ

PolySpace

- 従来のテストでは基本的にほとんどオレンジ
  「テストではバグがあることは示せるが、バグがないことは証明できない」（ダイクストラ）
- PolySpace
  緑色の部分は実行時エラーは起こらない
- 赤は修正
- オレンジはレビュー

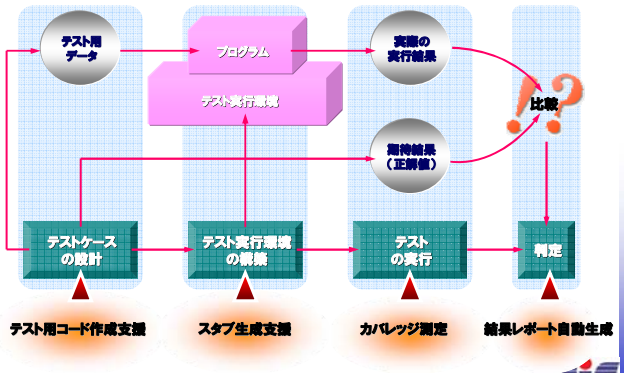
赤、とオレンジに集中、効率的にエラーをつぶす

コードレビュー工数の削減

Cantata++

Cantata++

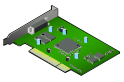
動的テスト

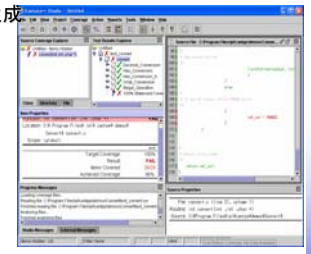


Cantata++

Cantata++

- 特徴
  - PC、シミュレータ、実機でのテスト
    - テストスクリプトは実装言語で生成
    - ステートメント、ブランチ、MC／DCカバレッジ

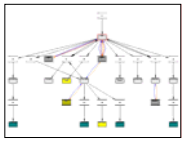




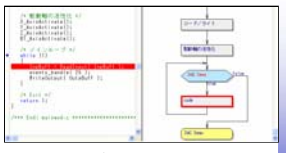
DAC（Development Assistant for C）

DAC

- レビュー支援ツール
- 機能
  - コード視覚化
    - フローチャート
    - 関数呼び出し図（コールツリー）
    - データフロー図
    - 型構造図
  - MISRA C 1998,2004対応の準拠性チェッカー
  - コーディングルール定義・チェック機能
  - メトリクス収集・グラフ化
  - ドキュメント生成
- 特徴
  - 図とソースコードとの対応



関数呼び出し図（コールツリー）



ソースコード　フローチャート

最後に

- 3月2日エーアイコーポレーションにて各ツール紹介セミナー開催
- ご清聴ありがとうございました！
- http://www.aicp.co.jp