『 組込みソフトウェア 品質向上ソリューションご紹介 』

株式会社エーアイコーポレーション
組込みテスト事業部
菅野 修也

株式会社エーアイコーポレーション

ツールによるテストの効率化、自動化

ソフトウェア規模・
複雑さの増大

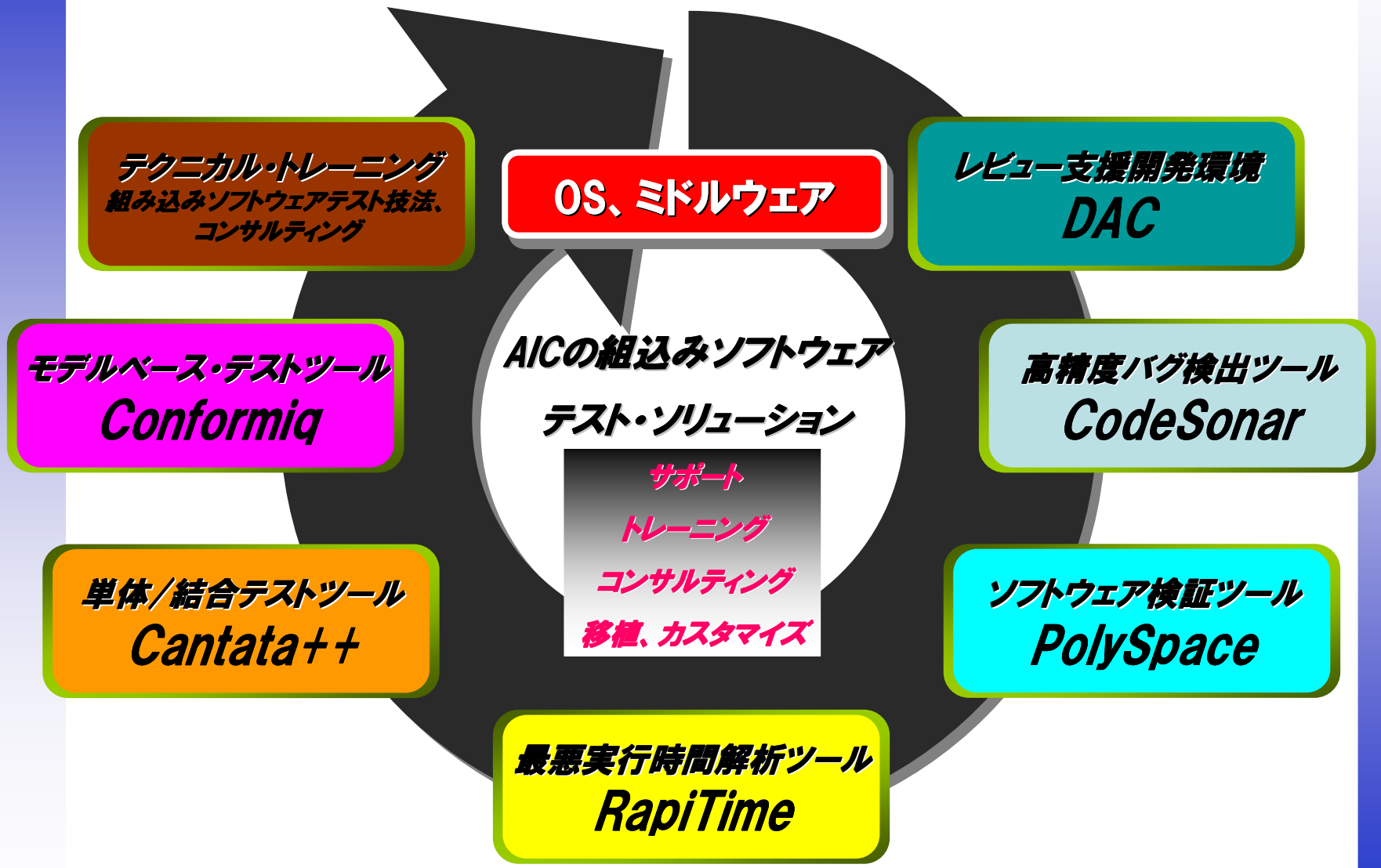
短納期化による
開発期間短縮

テストを効率的に行い、品質を確保する必要がある

人手だけではなくツールによる効率化！

- ヒューマンエラーをなくす
- テストの効率化、自動化

AIC 組込みソフトウェアテスト・ソリューション



高精度バグ検出ツール
「CodeSonar」
～メーカーとその解析技術～

はじめに

- GrammaTech社のCodeSonarは、C/C++で書かれたソースコードをコンパイル時に、深く解析し、様々な種類の重大なバグとセキュリティ上の脆弱性を検出し、ソフトウェアの品質を向上させる

GrammaTech社の沿革

- 本社は、米国ニューヨーク州のイサカ
  支社は、カルフォルニア州のサンホゼ

GRAMMATECH

- プログラミング言語、プログラム解析の分野において、PhDを取得している9人の研究者が在籍する
- GrammaTech社の『 CodeSonar 』は、現在米国コーネル大学のコンピュータサイエンス部のメンバーでもあるTim Teitelbaum氏の長年の研究により開発された、高精度バグ検出ツールである
- GrammaTech社の静的解析ツールは、米国のFortune500社から、教育研究所、政府関連企業など、世界のあらゆる分野において、使用されている

CodeSonarのコード解析技術の特長

- コンパイル時に解析を行い、不具合を検出
- プロシージャ間解析手法

- 検出した不具合を分析し、詳細にレポート
- 高いスケーラビリティ
- 煩わしい設定なしで、すぐに使用できる
- ソースコードやビルドシステムを変更する必要がなく、解析を実行できる

CodeSonarによるソリューション

- CodeSonarが検出できる不具合
  - ① ソフトウェア欠陥
    - » メモリリーク、NULLポインタ参照など
  - ② セキュリティ脆弱性
    - » バッファオーバーフロー、書式文字列の脆弱性など

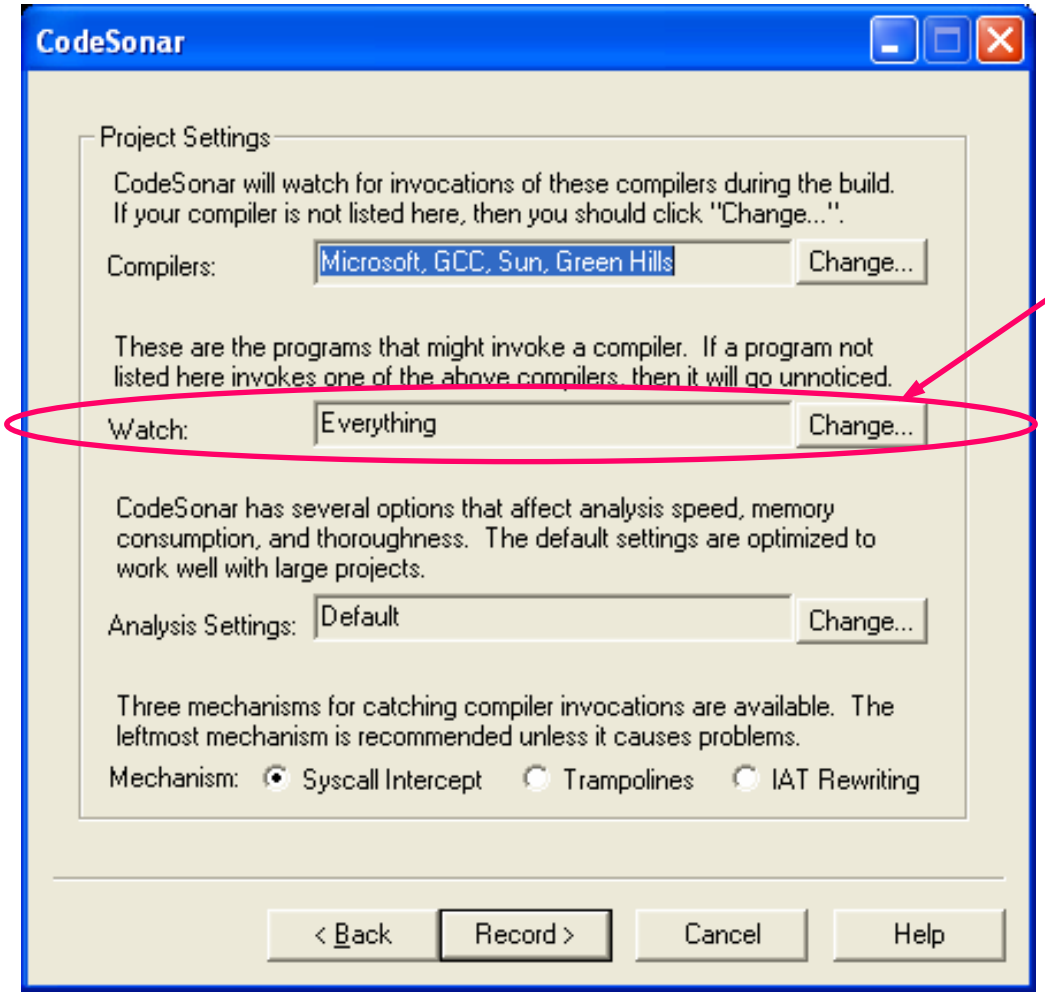
- 高精度バグ検出の原理（フォールスポジィティブを最小限に抑える）
  - ① ソースコード情報
    - » 不具合解析の対象
  - ② ビルド情報
    - » コードが実行しうる動作に関して、正確な情報を獲得し、コールグラフ、制御フローグラフなどを作成し、解析用のモデルを作成する

検出できる基本的なバグ

- バッファオーバラン
- バッファアンダーラン
- 危険な関数キャスト
- mallocで確保したオブジェクトを delete [] で解放
- newで確保したオブジェクトを delete [] で解放
- mallocで確保したオブジェクトを deleteで解放
- new [] で確保したオブジェクトを deleteで解放
- ゼロによる割り算
- 二重解放
- フォーマットストリング(書式指定文字列の脆弱性)
- 非ヒープ変数解放
- NULLポインタ解放
- newで確保したでオブジェクトを freeで解放
- 無視された戻り値
- メモリリーク
- リターン命令ミス(リターン命令がない)
- NULLポインタ参照
- 参照後のNULLテスト
- 冗長な条件
- ローカル変数へのポインタ戻し
- 解放メモリへのポインタ戻し
- 型オーバラン
- 型アンダーラン
- 未初期化変数
- 未到達コード
- 未使用値
- 解放後の使用
- 無意味な割当て
- 可変長引数関数キャスト
- キャストによる値変更
- ビット幅を超えるシフト
- 負の値のシフト

CodeSonarの環境設定

- 煩わしい設定なしで、すぐに使用できる



ビルドプロセスを
Watchする

開発者に有益な不具合レポート

Null Pointer Dereference

class of bug

Project Details	
Project:	BasicProj
User:	username
Date:	Tue Jan 10 16:07:57 2006
Machine:	FRED
Machine Type:	Ms-Win32
Bug Details	
Bug ID	1
File	your\working\directory\BasicNullDeref.c
Procedure	main
Program Point	buf[0] = *(q + 0)
Start Procedure	main
Path Length	7
Suppression	^Null Pointer Dereference: main / BasicNullDeref.c:21$
Other Bugs	↑Index | Next →

general project details

Legend
Buggy
Contributes
Two or More Loop Iterations
On Execution Path
Comment
Macro
Preprocessor
Include
Keyword
Preprocessed Away

description of syntax coloring

individual bug details

Problem	Line	Source
		your\working\directory\BasicNullDeref.c
		Enter main
	10	int main()
	11	{
	12	char buf[10];
	13	char *q;
	14	
	15	switch(rand())
	16	{
	17	case 1:
	18	/* a straightforward null dereference */
	19	
true	20	q = NULL;
q <= 4095	21	buf[0] = q[0]; /* Null Pointer Derefere
+ Preconditions for bug		
+ Postconditions for bug		

source file name

source file excerpt
with interesting lines highlighted

line numbers for source file

annotations indicating problem conditions

bug pre- and post-conditions

Generated on Wed Dec 07 11:42:47 2005 by GrammaTech CodeSonar 1.0p1 which was built on Dec 7 2005 00

CodeSonarの不具合レポート

検出例：NULLポインタ参照

Legend

- Buggy
- Contributes
- Two or More Loop Iterations On Execution Path
- Comment
- Macro
- *Preprocessor*
- Include
- **Keyword**
- *Preprocessed Away*

```
10  int main()
11  {
12      char buf[10];
13      char *p, *q;
14
15      switch( rand() )
16      {
17      case 1:
18          /* a straightforward null dereference */
19
20          q = NULL;
21          buf[0] = q[0];        /* Null Pointer Dereference (ID: 5) */
```

true
q <= 4095

- Preconditions for bug
  - $input_12 = 1
- Postconditions for bug
  - q' = 0

まとめ

- CodeSonarは、精度の高い解析機能を誇り、通常検出困難な不具合(実行時エラー、セキュリティ脆弱性など)を徹底的に検出する
  - 開発者の品質に対する問題解決を支援し、高い投資効果(ROI)が得られる
  - 検出される不具合は、本物である可能性が高い
- パス解析、データフロー解析を行い、レポートとして表示するので、不具合情報をより詳細に分析できる
  - レポートが見やすく、短時間で不具合の原因がわかる
  - 開発者の解析結果レビューにかける時間の無駄を省き、実際の不具合修正に集中ができる

ご清聴頂きまして、誠にありがとうございました。

株式会社エーアイコーポレーション
組込みテスト事業部
菅野 修也