動作環境

■システム構成　バージョン：C++test 2020.2

※2020.2で新規対応した環境が緑字です。

CPU		2.0GHz以上、マルチCPU構成を推奨
メモリ		4GB以上　(推奨：8GB以上)

Windows	OS	Windows 7 (32bit/64bit) Windows 10 (32bit/64bit) Windows Server 2008 (64bit) Windows Server 2012 (64bit) Windows Server 2016 (64bit)
	ホスト コンパイラ	•Visual C++ 9.0 (Visual Studio 2008)、10.0 (Visual Studio 2010)、11.0 (Visual Studio 2012)、12.0 (Visual Studio 2013)、14.0 (Visual Studio 2015)、14.1x (Visual Studio 2017)、14.2 (Visual Studio 2019) •GNU gcc/g++ x86：4.0.x、4.1.x、4.2.x、4.3.x、4.4.x、4.5.x、4.6.x、4.7.x、4.8.x、4.9.x、5.x、6.x、7.x、8.x、9.x •GNU gcc/g++ x86-64：4.8.x、4.9.x、5.x、6.x、7.x、8.x、9.x •National Instruments LabWindows/CVI 2013 Clang C/C++ Compiler v2.9 for Win32 (静的解析のみ) •National Instruments LabWindows/CVI 2015 Clang C/C++ Compiler v3.3 for Win32 (静的解析のみ) •National Instruments LabWindows/CVI 9.0 (静的解析のみ)
Linux (32bit)	OS	Linuxカーネル2.4以上、glibc 2.12以上、 x86互換プロセッサ
	ホスト コンパイラ	•GNU gcc/g++ x86：4.0.x、4.1.x、4.2.x、4.3.x、4.4.x、4.5.x、4.6.x、4.7.x、4.8.x、4.9.x、5.x、6.x、7.x、8.x、9.x •Clang C/C++ Compiler v3.9、v4.0、v5.0、v6.0 •Intel® C++ Compiler v18.0 •Green Hills MULTI for Linux x86 Native v4.2.x
Linux (64bit)	OS	Linuxカーネル2.6以上、glibc 2.12以上、 x86_64互換プロセッサ
	ホスト コンパイラ	•GNU gcc/g++ 4.0.x、4.1.x、4.2.x、4.3.x、4.4.x、4.5.x、4.6.x、4.7.x、4.8.x、4.9.x、5.x、6.x、7.x、8.x、9.x、10.x •Clang C/C++ Compiler v3.9、v4.0、v5.0、v6.0、v8.0、v10.0 •Intel® C++ Compiler v 18.0 •Green Hills MULTI for Linux x86 Native v4.2.x

■C++testをプラグインできる統合開発環境

- Eclipse for C/C++ Developers 4.2 – 4.15 (2020-03)
- Visual Studio 2012、2013、2015、2017、2019
- Wind River Workbench 4.0
- Arm Development Studio 5　(DS-5)　v5.28以上
- QNX Software Development Platform 7.0
- Texas Instruments Code Composer Studio IDE v7.4、v8.0

■共通項目：クロスコンパイラ

Renesas	Windows	•SuperH RISC engine C/C++ Compiler V5.1 (静的解析のみ)、V9.03 (C++については限定サポート)、V9.04 (C++については限定サポート) •M16C/R8C C Compiler 5.4x (静的解析のみ) •RX C/C++ Compiler 2.2x、2.5x ※下記のコンパイラについてはお問い合わせ先までご連絡ください。 •R32C/100 Series C Compiler •RH850 Family Compiler •RL78 (78K0R) C Compiler •RL78 (CC-RL) Family Compiler •H8S, H8/300 SERIES C/C++ Compiler (静的解析のみ) •V850 Optimizing C Compiler (静的解析のみ) •M32R Family C/C++ Compiler (静的解析のみ) •78K0 C Compiler (静的解析のみ)
Cypress Semiconductor	Windows	•FR Family SOFTUNE C/C++ Compiler V6
Green Hills Software	Windows	•Green Hills Software Compiler for PPC v3.5、v4.0.x、v4.2.x、v5.0.x、v2013.1.x、v2017.1.x •Green Hills Software Compiler for V850 v5.1.x、v2013.5.x、v2014.1.x、v2017.5.x •Green Hills Software Compiler for RH850 v5.1.x、v2013.5.x、v2014.1.x、v2017.5.x •Green Hills Software Compiler for ARM v2014.1.x、v2017.5.x (静的解析のみ) •Green Hills Software Compiler for ARM64 v2014.1.x、v2017.5.x (静的解析のみ)
	Linux	•Green Hills Software Compiler for V850 v2013.5.x、v2014.1.x、v2017.5.x •Green Hills Software Compiler for RH850 v2013.5.x、v2014.1.x、v2017.5.x •Green Hills Software Compiler for ARM v2014.1.x、v2017.5.x (静的解析のみ) •Green Hills Software Compiler for ARM64 v2014.1.x、v2017.5.x (静的解析のみ)
Arm	Windows	•Arm RealView 4.1 •Arm Compiler 5.0、6.6、6.9 •Arm GNU GCC 4.5.x
	Linux	•Arm Compiler 5.0、6.6、6.9、6.14
Texas Instruments	Windows	•TI TMS320C6x C/C++ Compiler v7.3、v7.4、v8.2 •TI TMS320C2000 C/C++ Compiler v6.2、v16.9、v18.1 •TI MSP430 C/C++ Compiler v4.0、v18.1 •TI MSP430 C/C++ Compiler GNU GCC 6.x •TI ARM C/C++ Compiler v5.1.x、v18.1 •TI ARM C/C++ Compiler GNU GCC 7.x
	Linux	•TI TMS320C6x C/C++ Compiler v7.3、v8.2 •TI TMS320C2000 C/C++ Compiler v16.9、v18.1 •TI MSP430 C/C++ Compiler v18.1 •TI MSP430 C/C++ Compiler GNU GCC 6.x •TI ARM C/C++ Compiler v5.1.x、v18.1 •TI ARM C/C++ Compiler GNU GCC 7.x
Wind River	Windows	•Wind River GCC 3.3.x、3.4.x、4.1.x、4.3.x、4.8.x •Wind River DIAB 5.7.x – 5.9.x •Wind River Clang 8.0.x
	Linux	•Wind River GCC 3.4.x、4.1.x、4.3.x、4.8.x •Wind River DIAB 5.7.x – 5.9.x
Cosmic Software	Windows	•Cosmic Software 68HC08 C Cross Compiler v4.6.x (静的解析のみ)
HighTec	Windows	•GCC for Tricore 4.9.x
IAR	Windows	•IAR Compiler for ARM v6.1x (C言語のみ)、v6.3x (C言語のみ)、v6.6x、v7.4x、v7.8x、v8.11.x、v8.22、v8.40、v8.50 •IAR Compiler for MSP430 v5.4x (C++ 言語は静的解析のみ)、v6.1 (静的解析のみ) •IAR Compiler for RX v2.5x、v2.6、v3.10.x •IAR Compiler for STM8 v1.4x　(静的解析のみ) •IAR Compiler for RL78 v3.1x •IAR Compiler for M16C & R8C v3.5x　(静的解析のみ)
Keil	Windows	•Arm C/C++ Compiler RealView 4.1 for uVision •Arm C/C++ Compiler 5.0 for uVision •Keil C51 Compiler v8.x　(静的解析のみ) •Keil C166 7.0　(静的解析のみ)
QNX	Windows	•QNX GCC x86 4.2.x、4.4.x、5.x •QNX GCC 5.x •QNX GCC (ARM) 5.x •QNX GCC (ARM64) 5.x
	Linux	•QNX GCC x86 5.x •QNX GCC 5.x •QNX GCC (ARM) 5.x •QNX GCC (ARM64) 5.x
Altium TASKING	Windows	•Altium TASKING VX-toolset for TriCore C/C++ Compiler 4.0、6.0、6.2、6.3 •Altium TASKING 80C196 C Compiler v6.0 r1 (静的解析のみ) •Altium TASKING classic compiler for C166/ST10 v6.0　(静的解析のみ)
Freescale	Windows	•Freescale CodeWarrior ANSI-C/cC++ Compiler 5.0.x for HC12 (静的解析のみ) •Freescale CodeWarrior C/C++ Compiler v.6.0 for ColdFire (静的解析のみ) •Freescale C/C++ Compiler v5.1 for Embedded ARM　(静的解析のみ)
Microchip	Windows	•MPLAB C30 Compiler for dsPIC v3.2x (静的解析のみ) •MPLAB C32 Compiler for PIC32 v2.0x (静的解析のみ)
National Instruments	Windows	•LabWindows/CVI 2013 Clang C/C++ Compiler v2.9 for Win32 (静的解析のみ) •LabWindows/CVI 2015 Clang C/C++ Compiler v3.3 for Win32 (静的解析のみ) •LabWindows/CVI 9.0 (静的解析のみ)
Mentor Graphics	Windows/Linux	•CodeSourcery Sourcery G++ Lite 2009q1-203 (静的解析のみ)
Embarcadero	Windows	•Embarcadero C++ Compiler 6.2x、6.9x for Win32 (静的解析のみ)
DesignWare ARC MetaWare	Windows/Linux	•Metaware DesignWare ARC C/C++ Compiler P-2019.09

※ GCC ベースのクロスコンパイラについてはお問い合わせください。

【開発元】

PARASOFT®

C++testの情報は　www.techmatrix.co.jp/product/ctest/

【総販売代理店】

TechMatrix

テクマトリックス株式会社
ソフトウェアエンジニアリング事業部
〒108-8588　東京都港区三田3-11-24　国際興業三田第2ビル
TEL：03-4405-7853　FAX：03-6436-3553
URL：https://www.techmatrix.co.jp/
E-MAIL：parasoft-info@techmatrix.co.jp

PARASOFT®

Parasoft C++test

for IEC 61508
for ISO 26262
for IEC 62304

C言語/C++言語対応 静的解析・単体テストツール C++test

ソフトウェアの品質向上と効率的な開発の実現をサポート

- コーディング規約チェック
- フロー解析
- AI&機械学習
- 単体テスト
- カバレッジ計測
- カバレッジアドバイザー
- アプリケーションモニタリング
- 組み込みソフトウェア開発での利用
- CIツール連携
- Docker連携
- Google Test連携
- レポート生成
- 機能安全認証取得
- コンプライアンスパック

C++test 体験版

TechMatrix

C言語／C++言語対応 静的解析・単体テストツール

Parasoft C++test

MISRA、AUTOSAR、CERT、CWEなどのコーディング規約チェック、単体テスト、カバレッジの計測などさまざまな要件に対応

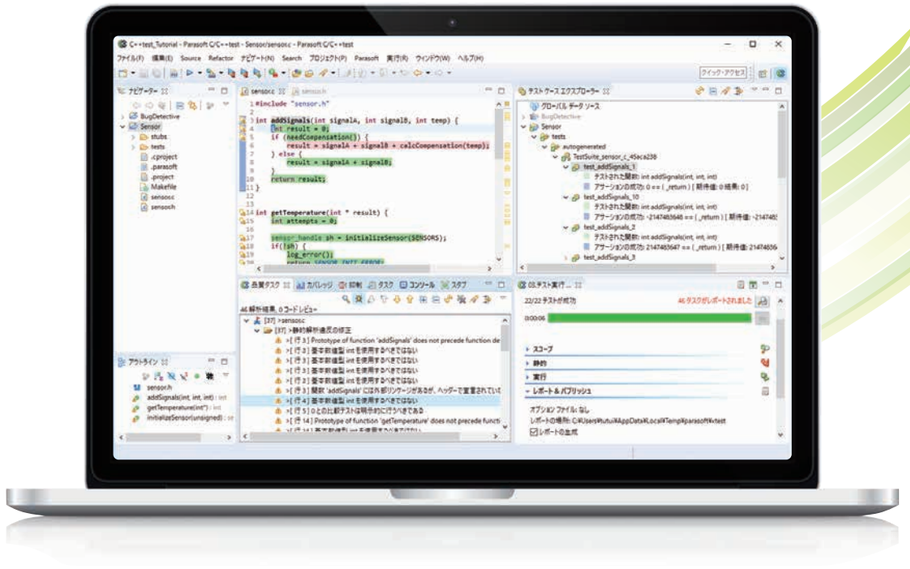
C++testは、静的解析(コーディング規約チェック／フロー解析)、単体テスト実行時メモリエラー検出機能などを搭載したC言語/C++言語対応のオールインワンテストツールです。ソフトウェア開発の工程に、C++testによる静的解析や単体テストを組み込むことにより、テストの効率化とソースコードの品質向上が期待できます。

PARASOFT × TechMatrix

導入後も安心してお使いいただけるサポート体制

C++test導入時のサポートから、運用支援、問題発生時のQ&A対応など、
導入後も安心してお使いいただけるサポート体制でお客様をバックアップいたします。

車載機器、産業機器、医療機器、OA機器といった組み込みソフトウェアの開発や、ミドルウェアの開発、Windows/Linuxアプリケーションの開発など、さまざまな現場で利用されています。



静的解析

**静的解析で致命的なバグを見逃さない！
AI&機械学習により解析結果の確認を支援**

コーディング規約チェック機能およびプログラムのあらゆるパスをシミュレートするフロー解析でバグを早期に発見します。
静的解析で検出された違反に対して修正するべきかどうかの予測を行う、独自のAIおよび機械学習 (ML) 機能を提供しています。

・コーディング規約チェック
・フロー解析
・AI&機械学習

動的解析

**テストドライバー・スタブ・テストケースの生成、
カバレッジアドバイザー機能で、単体テストを効率化**

GUI操作で「テストケース」の作成や「スタブ」の生成、スタブの振る舞いの設定ができます。
カバレッジを計測して単体テストの網羅性を視覚的にレポートします。また、効率的にカバレッジを向上させるためのテストデータ作成を支援します。

・単体テスト
・カバレッジ計測
・カバレッジアドバイザー
・アプリケーションモニタリング
・組み込みソフトウェア開発での利用

補助機能

効率的な運用や規格遵守を補助する機能を搭載

C++testは、第三者認証機関であるTÜV SÜD社よりIEC 61508およびISO 26262、IEC 62304に準拠したテストツールとして認証を取得済みです。また、CIツール連携、Docker連携、Google Test連携、レポート生成など各種機能を搭載しています。

・CI (継続的インテグレーション) ツール連携
・Docker連携
・Google Test連携
・レポート生成
・機能安全認証取得

コンプライアンスパック

**MISRA、AUTOSAR、CERT、CWEなどの
コーディング規約チェックに対応**

MISRA、AUTOSAR、CERT、CWEなどの遵守状況をリアルタイムに表示するダッシュボード画面の提供、コーディングガイドラインに則った遵守サマリーレポートや逸脱のレポートを自動生成します。

対応規格を一部抜粋
・MISRA C:2012 Amendment2
・AUTOSAR C++14コーディングガイドライン
・CERT C コーディングスタンダード
・CERT C++ コーディングスタンダード
・CWE/SANS TOP 25

1 静的解析

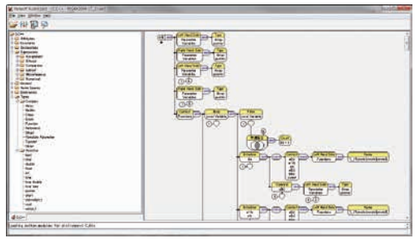
コーディング規約チェック

バグの作り込みを防止、ソースコードの可読性と保守性を強化

- ●コーディングルール4,000種類以上（フロー解析込み）を搭載
- ●MISRA、AUTOSAR、CERT、CWEなどコーディング規約に対応
- ●独自のルールセットの作成やコーディングルールの追加・編集
- ●メトリクス計測、重複コード検出機能を装備

MISRA C/C++、AUTOSAR C++14コーディングガイドライン、CERT C/C++コーディングスタンダード、CWE Top 25などの規約に対応しています。バグの作り込みを防止し、ソースコードの可読性と保守性、拡張性に優れた高品質で寿命の長いソースコードの実装を支援します。
また、ユーザー定義コーディングルールを作成する「RuleWizard」を搭載しています。社内やプロジェクトのコーディング規約に合わせて、独自のコーディングルールに修正したり、新規にルールセットを作成できます。他にも、メトリクス計測や重複コード検出機能を備えています。

●独自のコーディングルールの作成



●コーディングルールセット（抜粋）

- ・AUTOSAR C++14コーディングガイドライン
- ・MISRA C:1998 、MISRA C:2004、MISRA C++:2008、MISRA C:2012 規約チェックルール
- ・CERT C、CERT C++ チェックルール
- ・HIS ソースコードメトリクス チェックルール
- ・FDA C/C++ 推奨ルール
- ・OWASP TOP 10、PCIDSS、CWE/SANS 最も危険なプログラミングエラーTOP 25など、セキュリティに関するルール
- ・IPA/SEC コーディング作法ガイドチェックルール

フロー解析

プログラムのあらゆるパスをシミュレートし、バグを早期に発見

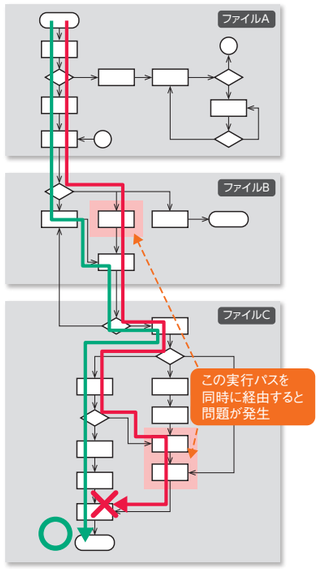
- ●フロー解析機能のルール数は117種類
- ●複数のファイル、関数にまたがる欠陥を自動的に検出

フロー解析は、プログラムを静的に解析して、プログラム実行時に発生し得る問題を検出します。複雑なアプリケーションでも、複数のファイル、メソッドにまたがるパスを自動的にトレースし、NULLポインターの間接参照やバッファオーバーフローなどプログラムの動作に致命的な影響をもたらすバグを早期に発見します。また、Parasoft DTPと連携することにより、フロー解析におけるデータフローのシミュレート結果をより詳しく表示できます。

●検出可能な項目（抜粋）

- ・メモリリーク/リソースリーク
- ・バッファオーバーフロー
- ・NULLポインターの参照
- ・未初期化変数の参照
- ・整数オーバーフロー
- ・ゼロ除算
- ・配列の境界外アクセス
- ・イテレーター範囲外アクセス
- ・セキュリティ脆弱性
- ・デッドロック
- ・不適切な排他制御

●ファイルをまたがる問題を検出（イメージ図）



AI&機械学習

優先して修正すべき違反をすぐに認識

- ●AI&機械学習による解析結果の確認を支援

静的解析で検出された違反に対して修正するべきかどうかの予測を行う、独自のAIおよび機械学習（ML）機能を提供しています。AIおよび機械学習（ML）機能は、過去の静的解析の違反に対するユーザーのアクションなど複数のデータを組み合わせた分析をツールが行い、静的解析の各違反に対して、修正するべきかどうかの予測を行います。

2 動的解析

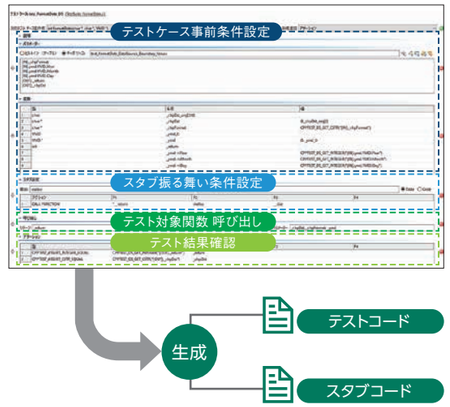
単体テスト

GUI操作でテストケースとスタブを生成。テストの実行と回帰テストを自動化

- ●テストドライバー、スタブ、テストケースを生成し、ソフトウェアの単体テストを自動化
- ●Excelで管理しているテストデータ、CppUnitのテストケースを活用

単体テスト時の課題であった「テストのためのコーディング」を行う必要はありません。C++testはGUI操作のみで「テストケース」の作成や「スタブ」の生成、さらにスタブの複雑な振る舞いの設定も可能です。また、テストケースとスタブを1つの画面でコントロールできるため、管理、メンテナンスを容易に行うことができます。
テストケース、スタブを作成するための工数およびこれらを管理、メンテナンスするための工数を大幅に削減します。また、外部テストデータの取り込みや、既存のテスト資産の再利用が可能です。

●テストケース・スタブを1画面でコントロール



カバレッジ計測

9種類のカバレッジを計測。単体テストの網羅性を視覚的にレポート

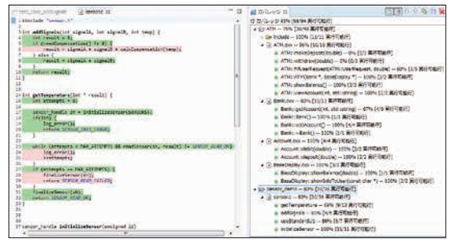
- ●プロジェクト、ファイル、関数単位でカバレッジの計測が可能
- ●実行/未実行の箇所をハイライト表示

単体テスト実行時に自動的に9種類のカバレッジを計測します。複数のカバレッジを同時に計測することもできます。画面上で実行/未実行の箇所を分かりやすくハイライト表示するため、視覚的に確認することができます。

●C++testがレポートするカバレッジ

- ・ステートメントカバレッジ（C0:命令網羅率）
- ・判断文カバレッジ（C1:分岐網羅率）
- ・単純条件カバレッジ（C2:条件網羅率）
- ・MC/DC（Modified Condition/Decision Coverage）
- ・関数カバレッジ
- ・コールカバレッジ
- ・行カバレッジ
- ・基本ブロックカバレッジ
- ・パスカバレッジ

●9種類のカバレッジを自動的に計測（行カバレッジの計測結果の例）



カバレッジアドバイザー

カバレッジアドバイザー（Coverage Advisor）で、単体テストを効率化

- ●効率的にカバレッジを向上させるためのテストデータ作成を支援
- ●実行できていない行をエディタで確認
- ●複雑な条件分岐も瞬時に計算

カバレッジアドバイザーは、単体テストにおいて誰もが抱える「中身が複雑なコードのカバレッジを満たすのが大変」、「テストケース作成に必要なパラメータ、事前条件が多くて洗い出すのが大変」これらの悩みの解決へアプローチします。
ソースコードの任意の行に対するワンステップの操作で、その行のカバレッジを満たすのに必要なテストのパラメータや事前条件を把握できます。事前条件を即座に把握できるため、ユーザーのテストに掛かる時間や労力を大幅に削減できます。

●カバレッジアドバイザーの実行結果イメージ

2 動的解析

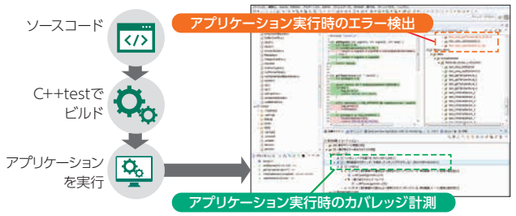
アプリケーションモニタリング

アプリケーション実行時に、メモリ関連エラーの検出とカバレッジを計測

●アプリケーション実行時のカバレッジを計測
●アプリケーション実行時に発生したエラーを自動検出

C++testは、システムテストを実施しながらカバレッジを計測することで、テストの抜け漏れを効率的に確認できます。また、不正メモリアクセス・メモリ破壊・メモリリーク・未初期化メモリの参照・NULLポインター参照などを検出し、スタックトレースと併せて問題をレポートします。また、システムテストに限らず、他のユニットテストフレームワークや独自のユニットテストフレームワークでのテスト実行時のカバレッジを計測できます。

●アプリケーション実行時のカバレッジ計測イメージ



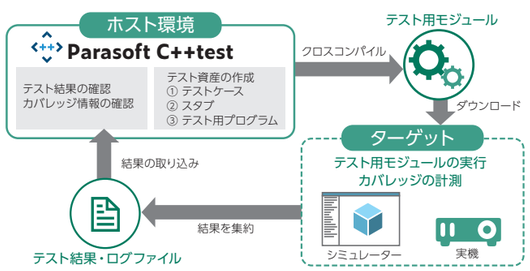
組み込みソフトウェア開発での利用

実機やシミュレーターで、単体テスト・カバレッジ計測が可能

●ホスト、シミュレーター、ターゲット環境で実行可能

C++testをインストールしたホストマシンだけでなく、実機（ターゲット機）や開発環境などに付属するシミュレーター上でも、単体テスト、カバレッジ計測（単体テスト時とアプリケーション実行時）および実行時メモリエラー検出を実行できます。C++testは、さまざまな組み込みソフトウェアのクロス開発環境をサポートしています。

●組み込みソフトウェアでの単体テスト実行イメージ

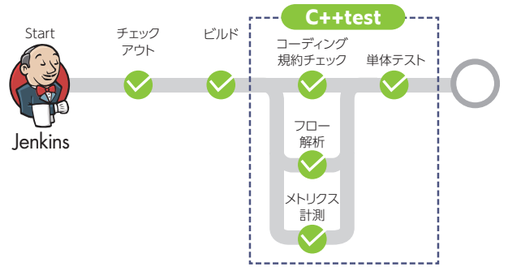


3 補助機能

CI（継続的インテグレーション）ツール連携

●静的解析および単体テスト実行プロセスを自動化

JenkinsなどのCIツールと連携することにより、静的解析および単体テストを定期的に自動実行することができます。
CIを導入することにより、開発者がコードをリポジトリにコミットするだけで、解析・テスト・レポートが自動化されるため、エラーや欠陥のフィードバックサイクルが早まり、品質の高いソフトウェアをより高速にリリースすることができます。



Docker連携

C++testは、Dockerコンテナでの作業をサポートしています。C/C++ツールチェーンまたはDockerコンテナによって提供されるテスト環境を使用して解析やテストを実行できます。

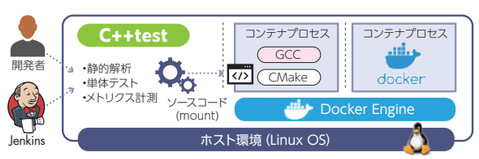
●Dockerコンテナ上のビルド環境を利用したテストが可能

ホスト環境上にインストールされたC++testから、Dockerコンテナ上に存在するビルド環境を利用し、静的解析および単体テストを実行することができます。Jenkinsからも実行可能なため、CI環境にも組み込むことができます。

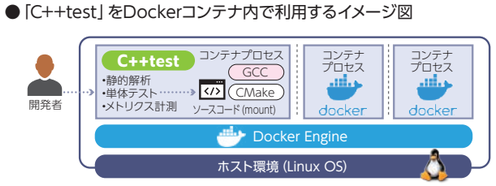
●Dockerイメージの配布で、テスト環境構築作業が不要に

C++testは、仮想環境上でも動作可能であるため、C++testを組み込んだ「Dockerイメージ」を開発者に配布することができます。これにより、ビルド環境だけでなく、C++testのテスト環境の構築作業もゼロにすることができます。

●「C++test」からDockerコンテナ内の環境を利用するイメージ図



●「C++test」をDockerコンテナ内で利用するイメージ図



Google Test連携

●Google Testでのテスト実行時のカバレッジ計測

Google Test で作成した既存のテストケースをそのままテスト資産として活用することができます。C++test のアプリケーションモニタリング機能を使用し、Google Test のテスト用プログラムを C++test を経由してビルドすることで、「C2 カバレッジ」、「MC/DC カバレッジ」など 9 種類のカバレッジを取得することが可能となります。

レポート生成

●豊富な情報を見やすいレイアウトでレポート出力

テスト結果をHTML、PDF、XML、CSV形式でレポートとして出力できます。コーディング規約が、守られていることを証明する場合などに利用できます。テスト実行に関する詳細な追加情報を出力することも可能です。

CSV　XML　PDF　HTML

機能安全認証取得

●IEC 61508 / ISO 26262 / IEC 62304準拠

C++testは、第三者認証機関であるTÜV SÜD社よりIEC 61508およびISO 26262、IEC 62304に準拠したテストツールとして認証を取得済みです。

《機能安全規格準拠に役立つルールセット》
・HISソースコードメトリクス チェックルール
・MISRA C:1998 、MISRA C:2004、MISRA C++:2008、MISRA C:2012 規約チェックルール

《医療機器ソフトウェア安全規格対応ルールセット》
・FDA C/C++（米国食品医薬品局）に関するルール

for IEC 61508
for ISO 26262
for IEC 62304

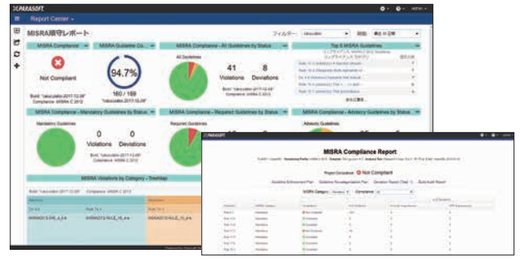
4 コンプライアンスパック

品質状況をリアルタイムに表示、レポートを自動生成

●MISRA、AUTOSAR、CERT、CWEなどの遵守状況をリアルタイムに表示
●MISRA Complianceレポートを自動生成

C++testのコンプライアンスパックは、MISRA C:2012、AUTOSAR C++14コーディングガイドライン、CERT C/C++コーディングスタンダード、CWEなどの遵守状況をリアルタイムに表示するダッシュボード画面の提供、コーディングガイドラインに則った遵守サマリーレポートや逸脱のレポートを自動生成します。コーディングガイドラインの遵守状況の説明責任を果たすことが容易になるだけでなく、未遵守箇所を早期に特定し必要な措置を講ずることにより、欠陥のあるソフトウェアに関連するビジネスリスクを排除することが可能になります。

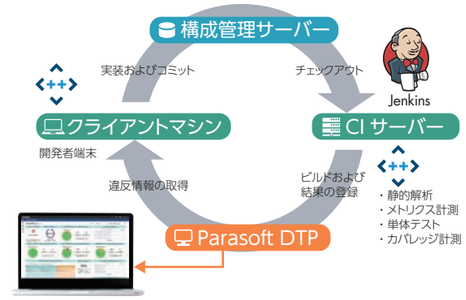
●MISRA C:2012 遵守用ダッシュボード画面・レポート出力



●Parasoft DTP（ダッシュボード）と連携したCI環境

Parasoft DTPは、C++testで行った静的解析/単体テストの結果、カバレッジなどの情報を自動的に収集・集約し、プロジェクトの状況をレポーティング、分析するためのツールです。
コーディング規約の遵守状況やプロジェクトの品質状況などをリアルタイムに表示します。CIに組み込むことで、常に最新のプロジェクト状況を確認することができます。開発者がレポーティングの作業をすることなく、管理者は各プロジェクトの状況を俯瞰して確認することができます。

●Parasoft DTPと連携したCI環境のイメージ図



最新の開発トレンドにいち早く対応！新たな言語規格/開発スタイルをサポート

C++testは、軽量なエディタであるVisual Studio Codeへのプラグインや、Dockerコンテナやクラウド環境での利用、Modern C++（C++17やC++20対応）をサポートしています。分散型SCMであるGitベースの開発ワークフローにシームレスに統合して利用することもできます。車載ソフトウェアを始めとして、組み込みソフトウェアでも採用が増えている開発スタイルにも適用できます。