



コンパイラの品質と機能安全

～ ユーザーがコンパイラをテストすべき理由 ～

コンパイラの品質:バグのないコンパイラはない

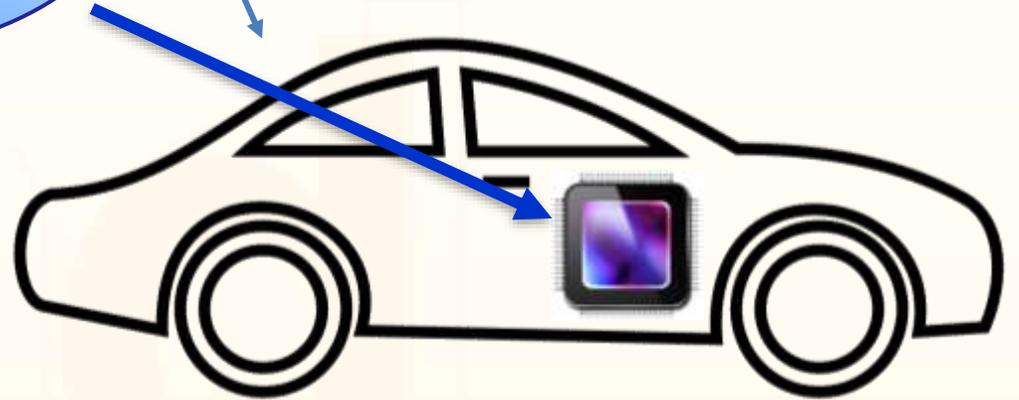


C/C++
ソースコード

コンパイラの欠陥は、コードの開発段階や、市場投入後に甚大な問題を引き起こし兼ねない

その損害は、コンパイラの品質管理への投資と比較にならない

C/C++コンパイラ



コンパイラ: テストも認証も難しい



- 巨大で非常に複雑 GCCを例にとると、
 - C, C++コンパイラのソースコードサイズ: 500万行
 - 開発期間: 40年近く、何千人ものエンジニアが関与
 - MISRA等への準拠やテストのカバレッジは実施されない
- MPUのバリエーションやオプション
 - 膨大な組合せ全てをメーカーはテストできない
- 開発はいつも進行中で修正、改善、機能追加など何らかの変更がある
 - 複雑なコンパイル過程での変更が、以降の処理の流れを全く違ったものに
 - バージョン3.4が信頼できるなら3.5も信頼できるということもない

コンパイラを認証することは現実的ではない

コンパイラは言語規格で評価できる



C++14 に対する G++7.3.0 のテスト結果サマリー

診断機能を評価



	LOG	
Compile errors	fail/chck/pass	
Language	81/	93/10250
Library	22/	0/2029
subtotal	103/93/12279	
Run errors	fail/chck/pass	
Language	4/	0/8026
Library	0/	0/1934
subtotal	4/ 0/9960	
Total	107/ 93/12275	

コード生成機能を評価



SuperTest™ : 言語規格に対するテストスイート



- C, C++言語規格への適合性、正確性、堅牢性をテストする
 - x 300万件以上のソースコード集
 - 30年以上ISO 言語標準規格に対応してきた経験と実績
- 開発に使用する固有のユースケースでコンパイラを評価
 - コード生成機能をオブジェクトコードの振る舞いで評価するテスト
 - 診断機能を評価するテスト
 - プログラムの構文や意味規則を診断
 - C,C++ 標準ライブラリのテスト



SuperTestの概要

テスト
ドライバ

テスト
レポータ

膨大な数の
手書きのテスト

言語仕様をカバー

T:コード生成機能の評価
X:診断機能の評価

莫大な数の
機械的に生成されるテスト

算術演算のテストなど

SuperTest ユーザ



主要なコンパイラメーカーやデバイスメーカーがコンパイラのテストやツール認定を目的に活用するなか、コンパイラユーザによる採用も増えています

コンパイラユーザがSuperTestで得た成果



- 同じコンパイラを1000本以上使用
 - バージョンアップごとにチェックし、使ってはいけない最適化などの社内ルールを施行。過去バージョンに戻すような問題が起こっていない
- ライブラリのテスト
 - C99にC89のライブラリが混在し数学関数で問題になるところを事前に回避
- Depthスイートテスト機能
 - アーキテクチャ固有のデータ長(例えば int が24ビット)に合わせて演算精度のテストができる
- 呼出し規約テスト機能でABI規則への準拠を確認
 - 他のサプライヤーの製品と繋がることによる問題の切り分け

コンパイラの品質を鵜呑みにできない

- 機能安全規格でソフトウェアツールチェーンの適正なレベルの信頼が要求される
- コンパイラの中身はユーザのコントロール外
- 固有のユースケースでコンパイラをテストして認定できるSuperTest
- コンパイラのテストも開発の早期段階で行うことが賢明！

C、C++コンパイラのツール認定の効用

https://www.fuji-setsu.co.jp/files/QualificationBenefitsSuperTest_JP.pdf

市場投入までの時間を加速

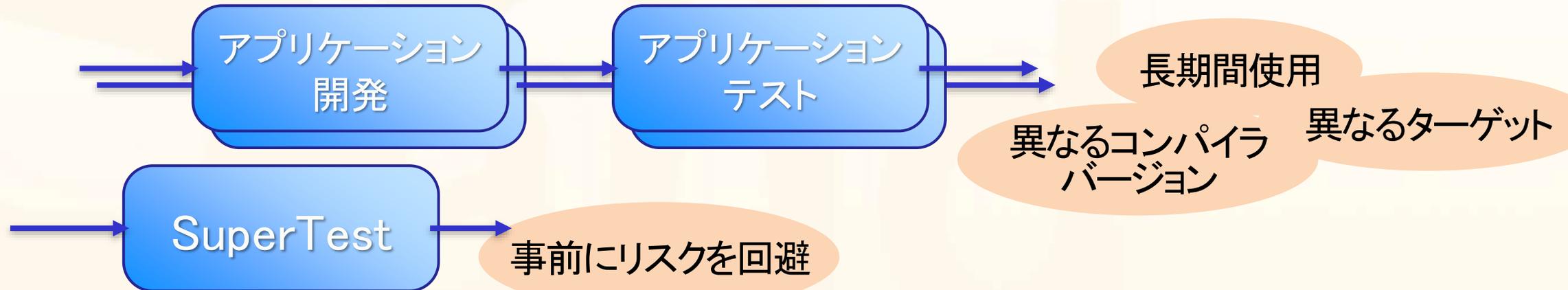


コンパイラテストなし: コンパイラの欠陥が後工程や出荷後に見つかり大きな手戻り発生



アプリケーションテストでコンパイラの全ての欠陥を発見することは非現実的

コンパイラテストあり: アプリ開発のプロセスに影響を与えるようなサプライズを回避できる

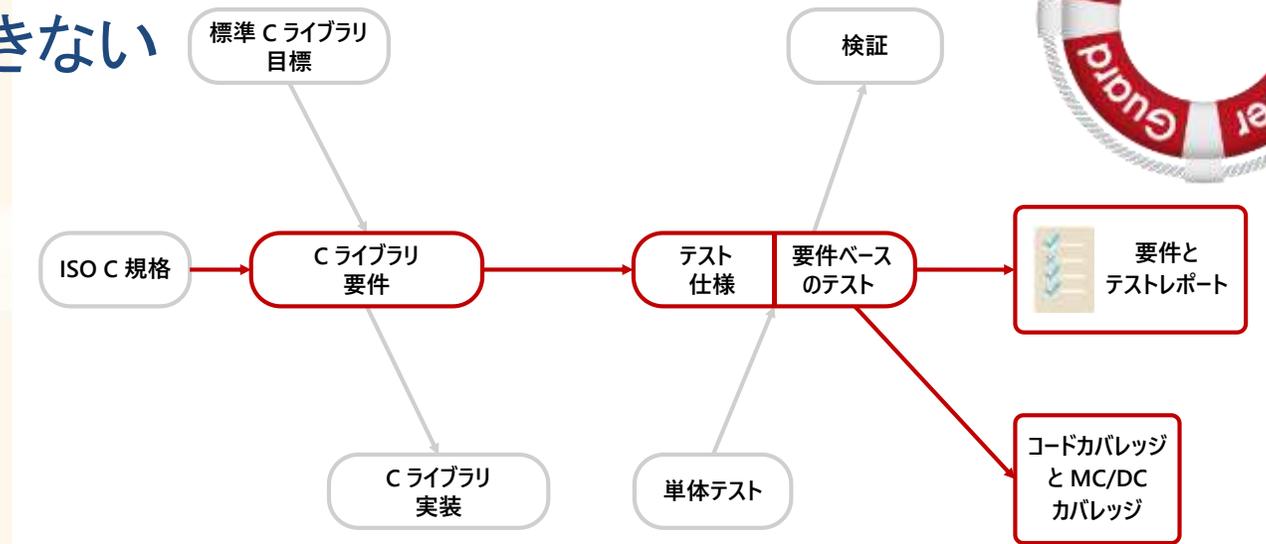


C,C++ 標準ライブラリ:コンパイラ認定とは別扱い

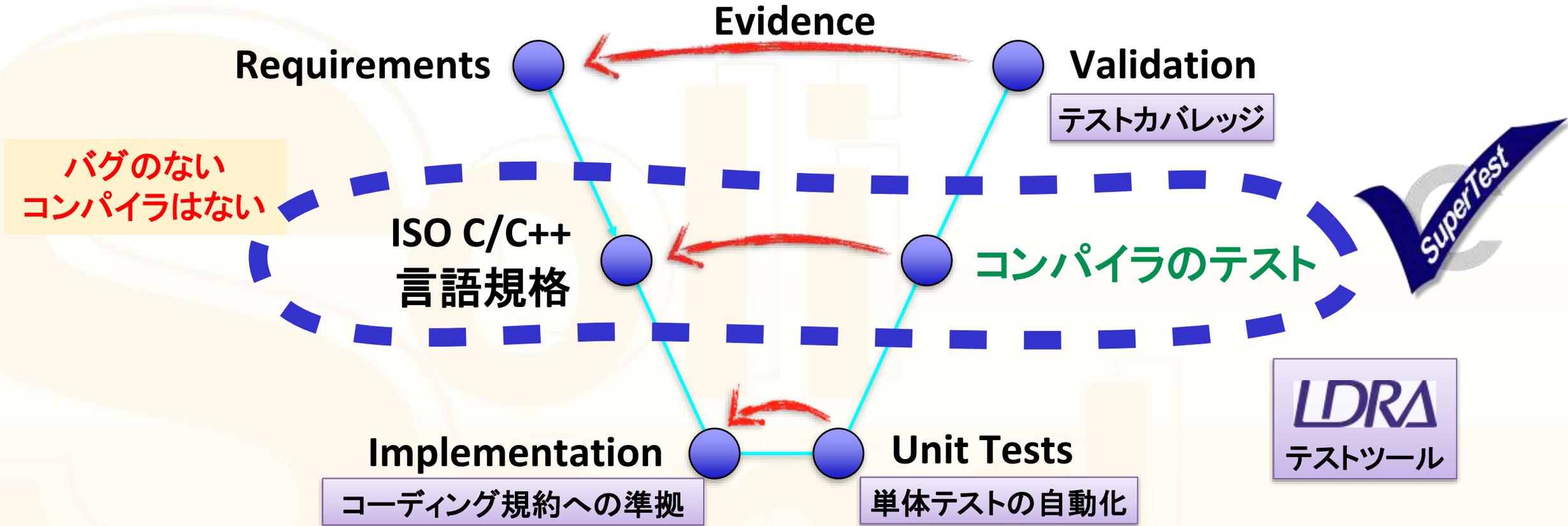


- ライブラリはアプリとリンクしてシステムの一部として実行される
- マクロやヘッダファイルなどソースコードに大きく依存するものもある
 - 言語仕様から導出される要件ベーステスト
 - 個別のコンパイラユースケース下で検証すべき
 - 事前認定されていても安心できない

- 要件の分析
- 同値クラスの使用
- 境界値の定義
- 「エラー推測」= 経験



検証作業とコンパイラ品質



The V-Model - Requirements Traceability



FUJI SETSUBI

Solid Sands社

<https://solidsands.com/>

SuperTest C/C++ コンパイラテスト

<https://www.fuji-setsu.co.jp/products/SuperTest/>



富士設備工業(株)電子機器事業部
<https://www.fuji-setsu.co.jp>