



Armのセキュリティと機能安全

つながる世界の
安心で安全なソフトウエアのテスト

富士設備工業(株)電子機器事業部

Security と Safety について

- 独語ではどちらも
  Sicherheit (ズィッヒャーハイト)

- 日本語では安心 や セキュリティ など、安全とは分けて表現されるが、モノがつながる時代に区別することはできない

- セキュリティの脆弱性が、
  機能安全上の潜在的な脅威になる

http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

様々な領域で現実的な問題に、、

Hackers are alleged to have destroyed a pump used to pipe water to thousands of homes in a US city in Illinois.

Hackers with access to the utility's network are thought to have broken the pump by turning it on and off quickly.

The FBI and Department for Homeland Security (DHS) are investigating the incident as details emerge of what could be a separate second attack.

The alleged attack was made on a system that piped clean water to homes in Illinois

上水道インフラ

http://www.bbc.co.uk/news/technology-15817335

Hack attack causes 'massive damage' at steel works

22 December 2014 | Technology

プラントの溶鉱炉

The hack attack led to failures in plant equipment and forced the fast shut down of a furnace

http://www.bbc.co.uk/news/technology-30575104

SF'S TRANSIT HACK COULD'VE BEEN WAY WORSE—AND CITIES MUST PREPARE

交通機関

https://www.wired.com/2016/11/sfs-transit-hack-couldve-way-worse-cities-must-prepare/

OPINION

Pacemaker hacker says worm could possibly 'commit mass murder'

ペースメーカ

It seems like something is very wrong with the picture when you read the news and it sounds more like a science fiction novel than a newsflash. For

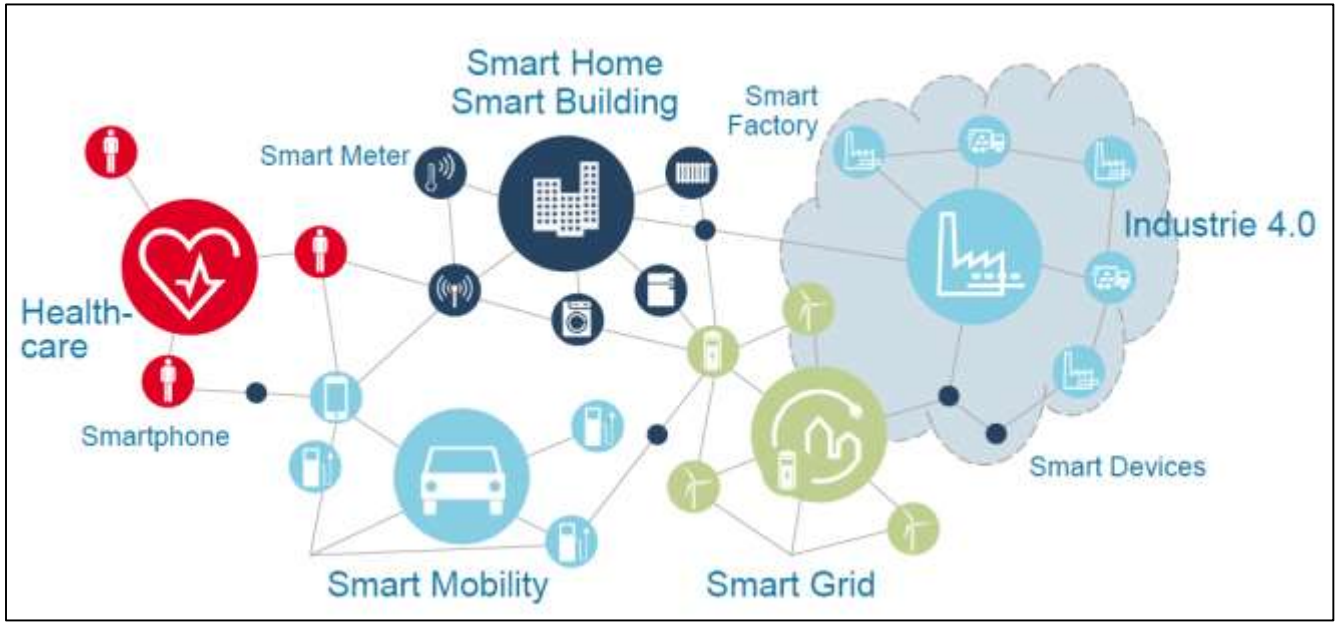
MORE LIKE THIS

Pacemaker hack can deliver deadly 830-volt

http://www.computerworld.com/article/2473402/cybercrime-hacking/pacemaker-hacker-says-worm-could-possibly--commit-mass-murder-.html#tk.drr_mlt

Internet of Things and Services

- 仮に機能安全が求められていない製品であっても、つながる世界で他への攻撃の踏み台にされるようではブランド価値の低下を免れない



https://www.informationweek.com/government/leadership/internet-of-things-8-cost-cutting-ideas-for-government/d/d-id/1113459

脆弱性とは

情報セキュリティ早期警戒パートナーシップガイドライン - IPA

- 脆弱性とは、ソフトウエア製品やウェブアプリケーション等において、コンピュータ不正アクセスやコンピュータウイルス等の攻撃により、その機能や性能を損なう原因となり得るセキュリティ上の問題箇所

https://www.ipa.go.jp/security/ciadr/partnership_guide.html

国民のための情報セキュリティサイト - 総務省

- 脆弱性とは、コンピュータのOSやソフトウェアにおいて、プログラムの不具合や設計上のミスが原因となって発生した情報セキュリティ上の欠陥のこと

http://www.soumu.go.jp/main_sosiki/joho_tsusin/security/basic/risk/11.html

Understanding Quality and Security

Wifi接続されるヘッドユニットへのパスワードを解読して侵入

*ヘッドユニットにつながるCANバス上のコントローラのファームウエアを変更（**権限が不要であった**）*

CANバス上のコンポーネントを制御下に

Thinking Safety vs Security

・高品質であるとしても必ずしもセキュアなソフトウェアとは言えない

・セキュアなソフトウェアは現在知られている あるいは将来の脅威への防御のために高品質でなければならない

・高品質のソフトウエアを構築するために必要な同じ開発ライフサイクルの厳格さを要求する => 脆弱性の原因となるプログラムの不具合や設計上のミスを排除

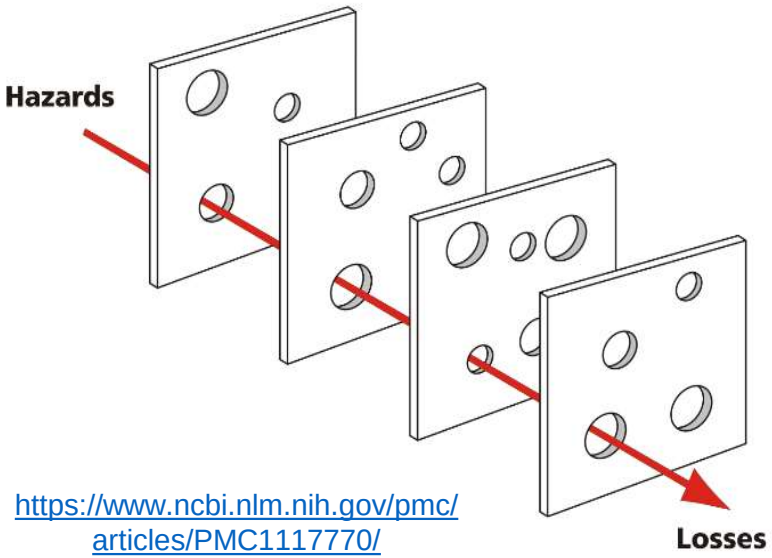
航空業界標準：LDRA社テストツール

- コーディングスタンダードチェック
- コード品質の尺度
- カバレッジ解析
- データフロー、コントロールフロー解析
- 単体テスト
  統合テスト
- 要件トレーサビリティ
- テストの管理
- 認証プロセスの支援

最も厳密なことで知られる航空業界のDO-178スタンダード認証に標準的に採用されるテストツールの各種機能が防御の一翼を担う

The Swiss Cheese model…

セキュリティへの単一解は無く複数の対策（多重防御）が施される



https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1117770/

スイスチーズモデル：事故は多重防護壁の穴をすべて貫通したときに生じる。穴は潜在的にも存在するし、即発的（**攻撃によって**）にも発生する。

事故を防ぐには穴の有無を常に監視し、発見したら直ぐに塞ぐ必要がある

- **セキュアブート**により正しいイメージがロードされる（不正な改ざんを検知）
- **ドメイン分離**でシステムのクリティカルな領域を保護
- **最小限の権限** (Least Privilege) **の設計原則**で脆弱性を最小限に抑える（Multiple Independent Levels of Securityのコアとなる原則）
- **攻撃対象領域を特定して最小限**に抑える
- セキュアコーディング標準
- セキュリティテスト

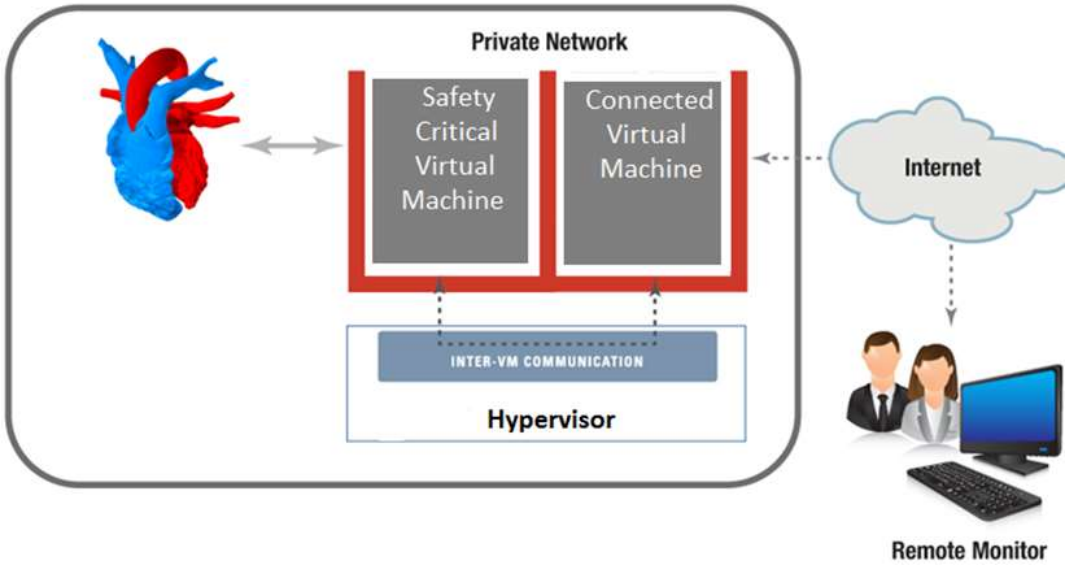
*アクセスが困難で脆弱性が最小限に抑えられるように、**コード自体が安全であることが重要***

And after release

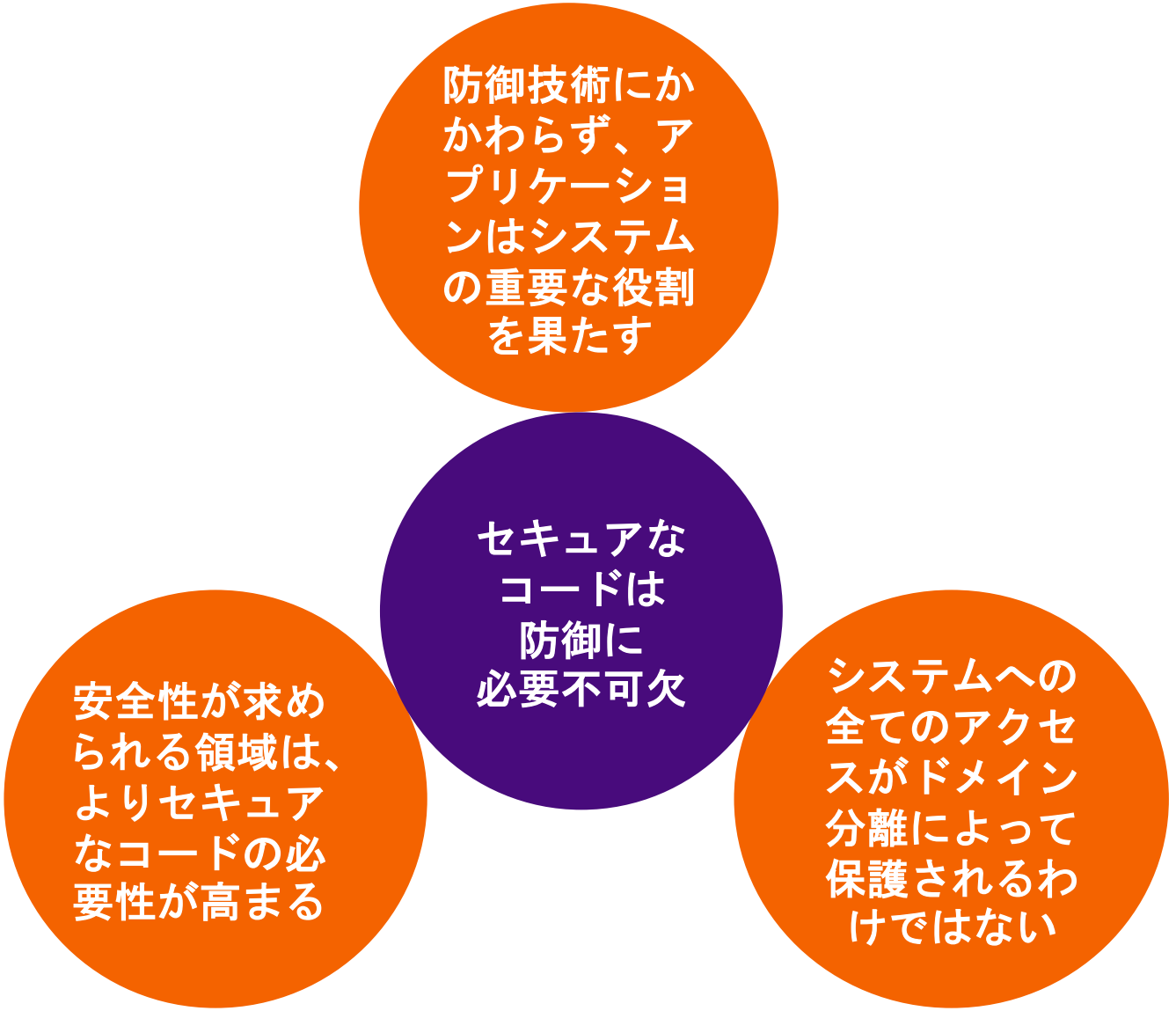
- 新たに発見される脆弱性に対処するためのレスポンシブなメンテナンスプロセス

Separation Technologies are just one slice of “Swiss Cheese”!

- ハイパーバイザーなど仮想化による分離は有効な防御策ではあるもののセキュリティを保証する完全策ではない
  - ハイパーバイザードメインの分離は、仮想化を実装するプロセッサー（IntelのVT-xやArmの仮想化拡張など）に制限される
  - DMAエンジンやGPUなど、システム内の他のバスマスタは、ハイパーバイザによって提供される保護をバイパスできる
  - 分離されるドメイン間の通信にアプリケーションコードが必要



Secure Application Code



Code Reviews

– Secure Coding

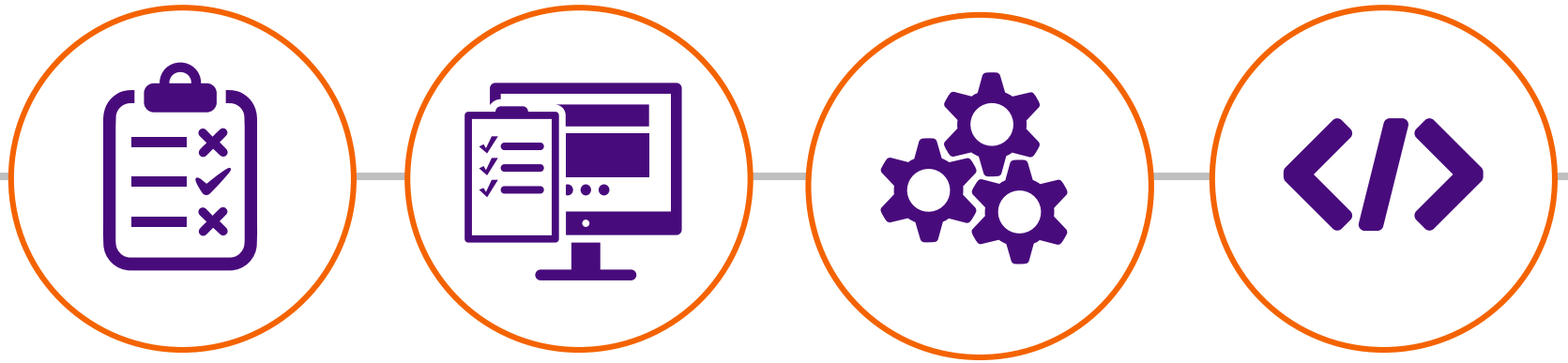
Coding Standards

ソフトウェアの欠陥の約80%は、言語の約20%の誤った使用によって引き起こされる

言語の使用を制限することで、問題のあることが分かっているサブセットを避けることができるなら、ソフトウェアの品質は大幅に向上する

処理系定義の動作

コーディングエラー



未規定の動作 / 未定義の動作

ランタイムエラー

Common Weakness Enumeration - CWE™

- 共通脆弱性タイプ一覧　https://cwe.mitre.org/

National Institute of Security Technologyの調査によると、ソフトウェアの脆弱性の64%はプログラミングエラーに由来する。 CWEプロジェクトは、ソフトウェアの欠陥をより深く理解し、その欠陥を特定、修正、防止するための自動ツールを作成することを目的とする。 CWEの共通脆弱性リストは、**米国国土安全保障省**の国家サイバーセキュリティ部門が共同で主催するソフトウェア保証戦略イニシアチブの一環として作成される。

共通脆弱性タイプ一覧CWE概説　https://www.ipa.go.jp/security/vuln/CWE.html

CERT® セキュアコーディング

- JPCERT/CCに日本語版公開

  https://www.jpcert.or.jp/sc-rules/

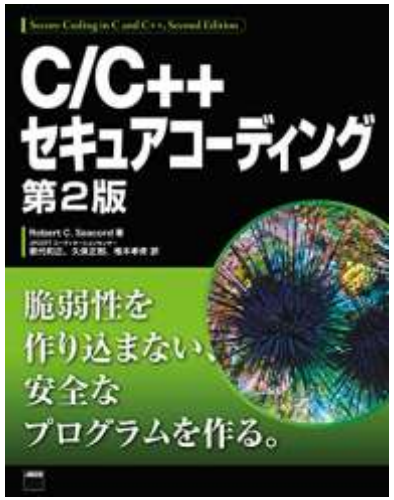
- CERT's Top 10 Secure Coding Practices

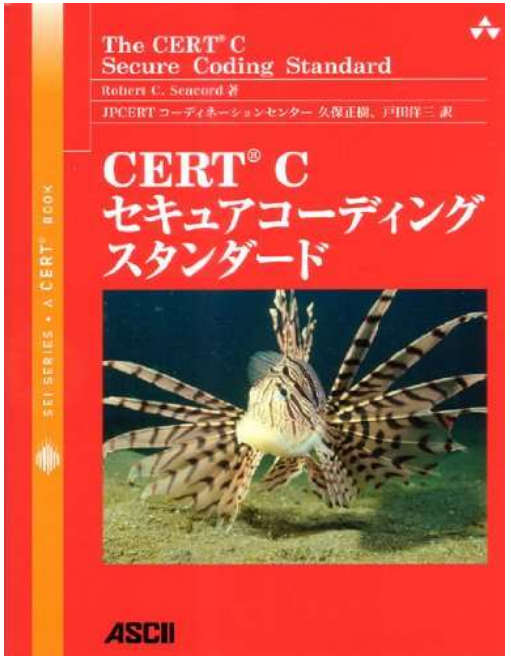
  IPAのセキュア･プログラミング講座（P12~）

  https://www.ipa.go.jp/files/000059838.pdf

- By Robert C. Seacord

DARPA(《米》国防総省国防高等研究事業局)が中心となってカーネギーメロン大学内に設置されたCERT/CC (Computer Emergency Response Team/Coordination Center) により公開





CERT | Software Engineering Institute Carnegie Mellon

MISRA C:2012 /AMD1/ADD2

MISRA C:2012 Addendum 2
ISO/IEC TS 17961:2013 の「Cセキュアコーディング」に対するMISRA C:2012のカバレッジ

x46 の C SecureルールのどれがMISRA C:2012のガイドラインの対象であるかを示す

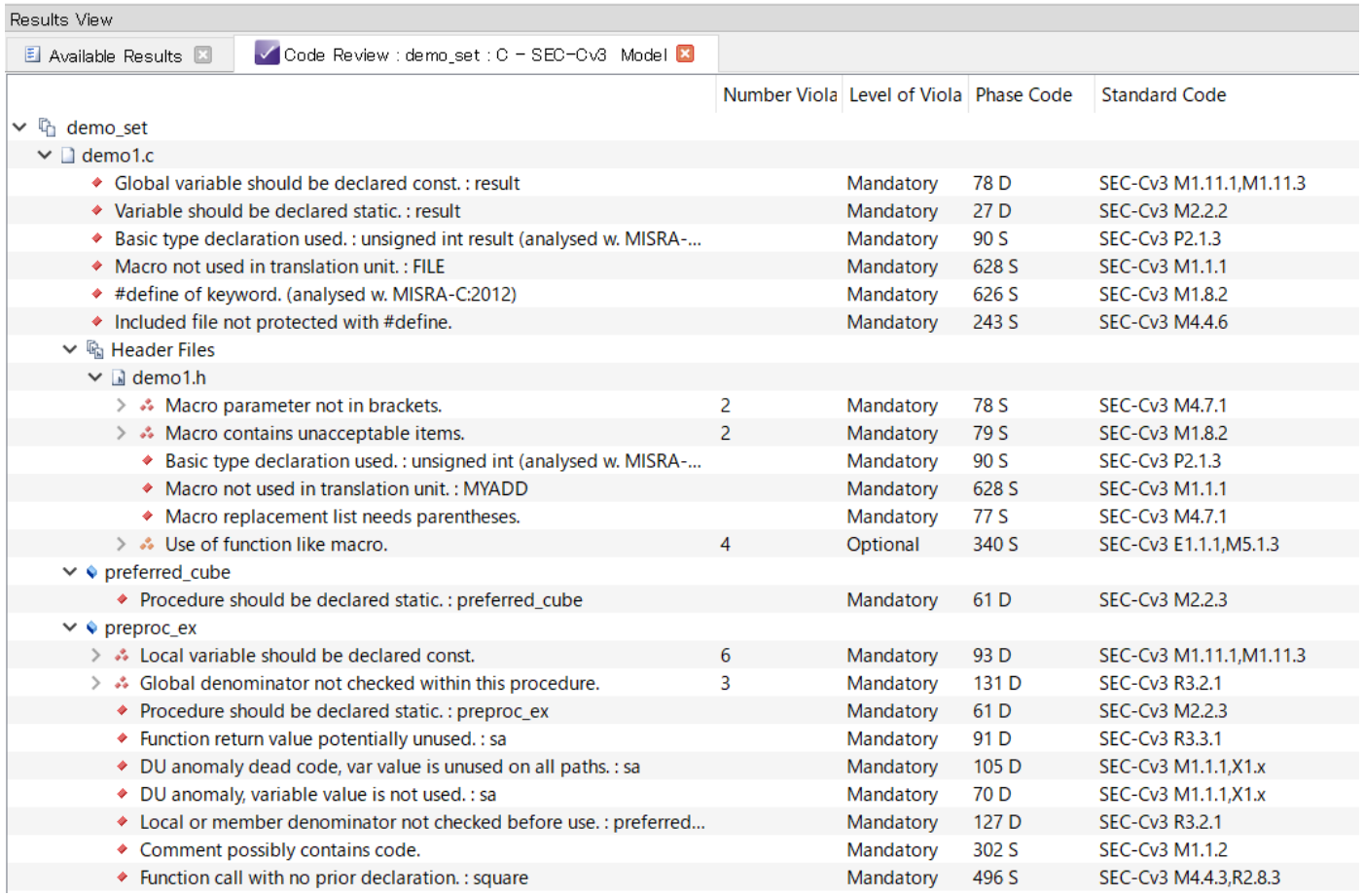
MISRA C:2012 Amendment 1
ISO C セキュアガイドラインへのカバレッジを強化する
14の新しいCコーディングルール

特に「信頼されていないデータの使用」というセキュリティ上の脆弱性に付き物の具体的な問題へ取り組む

セキュアでないコーディングの例
https://www.fuji-setsu.co.jp/files/MISRA_C_2012_AMD1.pdf

IPA ESCR

- IPA/SECの組込みソフトウェア開発向けコーディング作法ガイド ESCR［C言語版］Ver.3.0 の発刊
  ～セキュアコーディングへの対応～

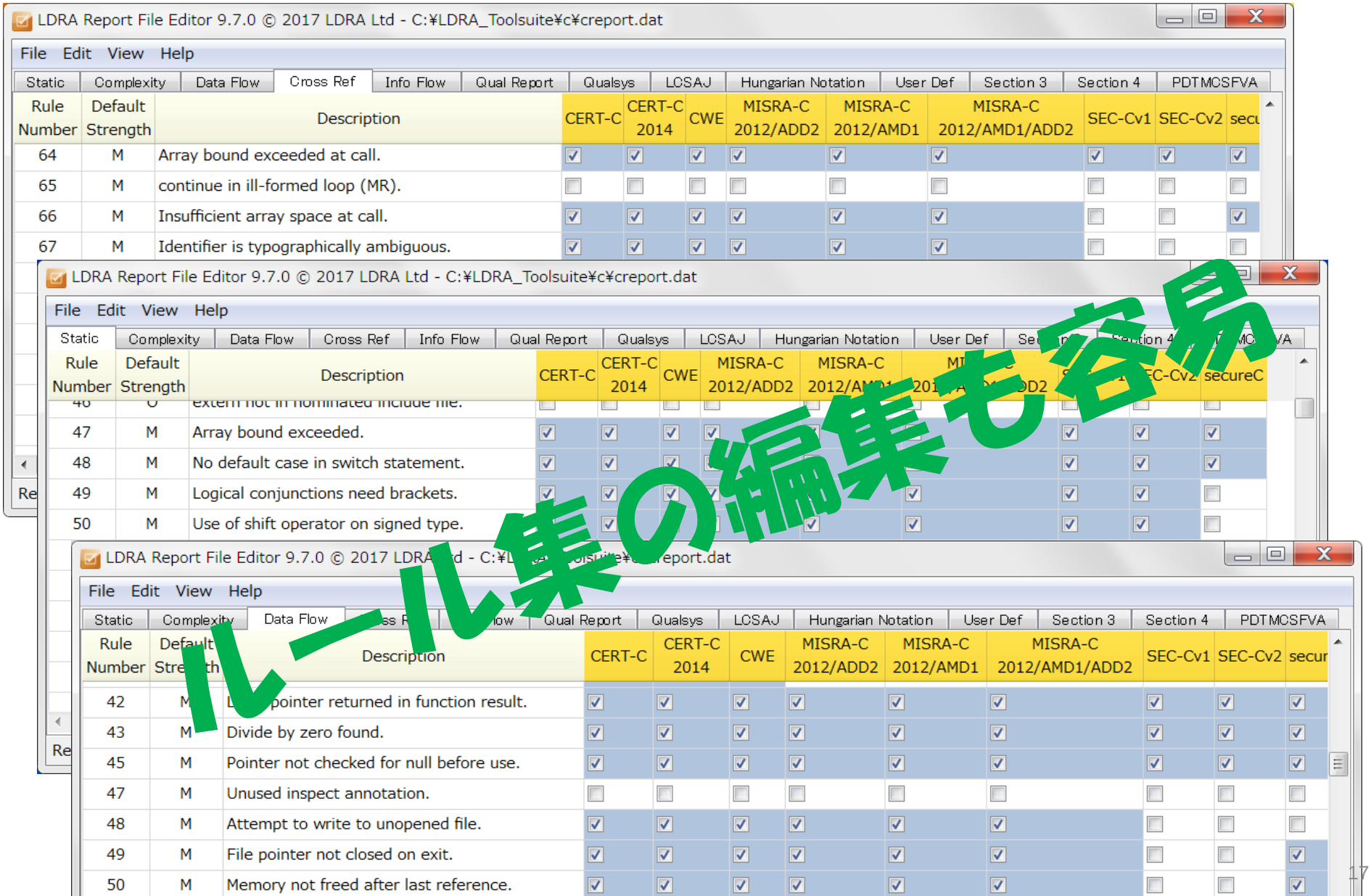


Results View

Available Results | Code Review : demo_set : C – SEC-Cv3 Model

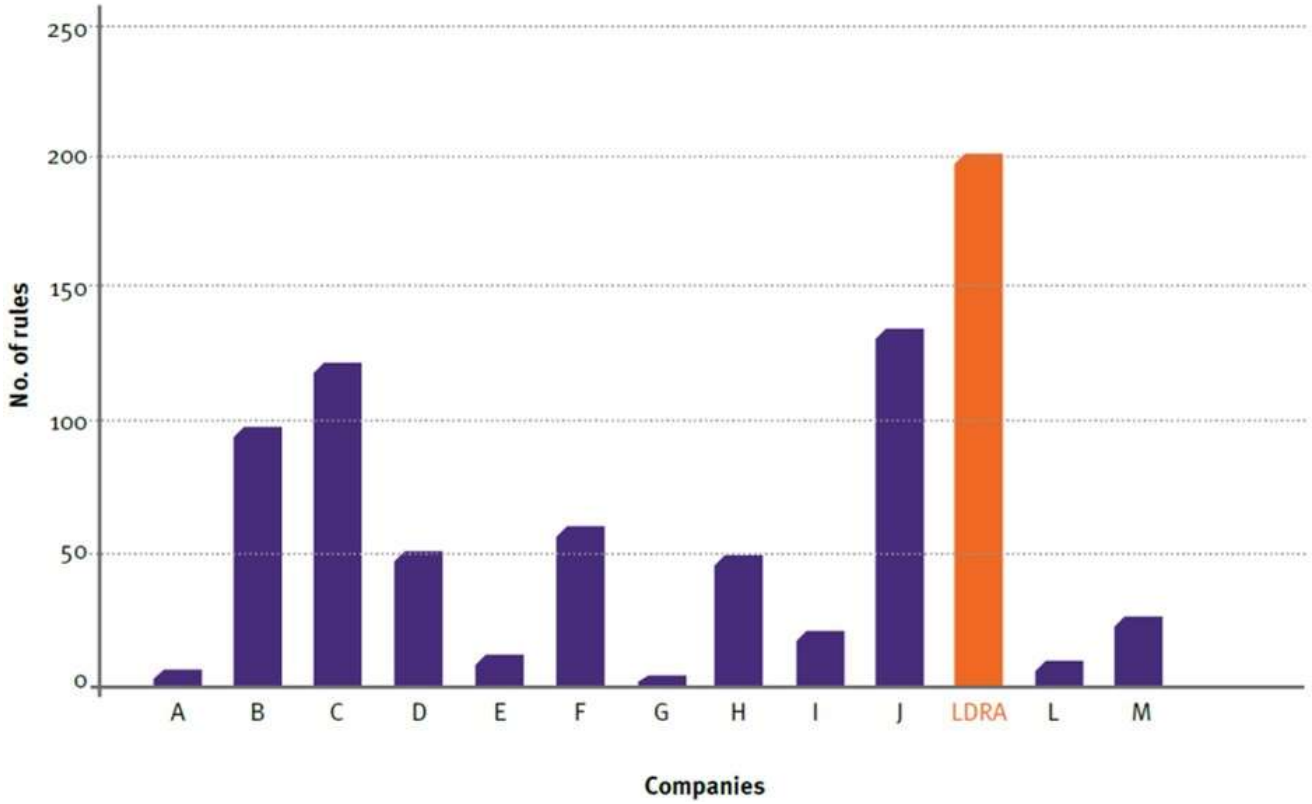
	Number Viola	Level of Viola	Phase Code	Standard Code
demo_set				
demo1.c				
Global variable should be declared const. : result		Mandatory	78 D	SEC-Cv3 M1.11.1,M1.11.3
Variable should be declared static. : result		Mandatory	27 D	SEC-Cv3 M2.2.2
Basic type declaration used. : unsigned int result (analysed w. MISRA-...		Mandatory	90 S	SEC-Cv3 P2.1.3
Macro not used in translation unit. : FILE		Mandatory	628 S	SEC-Cv3 M1.1.1
#define of keyword. (analysed w. MISRA-C:2012)		Mandatory	626 S	SEC-Cv3 M1.8.2
Included file not protected with #define.		Mandatory	243 S	SEC-Cv3 M4.4.6
Header Files				
demo1.h				
Macro parameter not in brackets.	2	Mandatory	78 S	SEC-Cv3 M4.7.1
Macro contains unacceptable items.	2	Mandatory	79 S	SEC-Cv3 M1.8.2
Basic type declaration used. : unsigned int (analysed w. MISRA-...		Mandatory	90 S	SEC-Cv3 P2.1.3
Macro not used in translation unit. : MYADD		Mandatory	628 S	SEC-Cv3 M1.1.1
Macro replacement list needs parentheses.		Mandatory	77 S	SEC-Cv3 M4.7.1
Use of function like macro.	4	Optional	340 S	SEC-Cv3 E1.1.1,M5.1.3
preferred_cube				
Procedure should be declared static. : preferred_cube		Mandatory	61 D	SEC-Cv3 M2.2.3
preproc_ex				
Local variable should be declared const.	6	Mandatory	93 D	SEC-Cv3 M1.11.1,M1.11.3
Global denominator not checked within this procedure.	3	Mandatory	131 D	SEC-Cv3 R3.2.1
Procedure should be declared static. : preproc_ex		Mandatory	61 D	SEC-Cv3 M2.2.3
Function return value potentially unused. : sa		Mandatory	91 D	SEC-Cv3 R3.3.1
DU anomaly dead code, var value is unused on all paths. : sa		Mandatory	105 D	SEC-Cv3 M1.1.1,X1.x
DU anomaly, variable value is not used. : sa		Mandatory	70 D	SEC-Cv3 M1.1.1,X1.x
Local or member denominator not checked before use. : preferred...		Mandatory	127 D	SEC-Cv3 R3.2.1
Comment possibly contains code.		Mandatory	302 S	SEC-Cv3 M1.1.2
Function call with no prior declaration. : square		Mandatory	496 S	SEC-Cv3 M4.4.3,R2.8.3

Coding Standards Models



ルール集の編集も容易

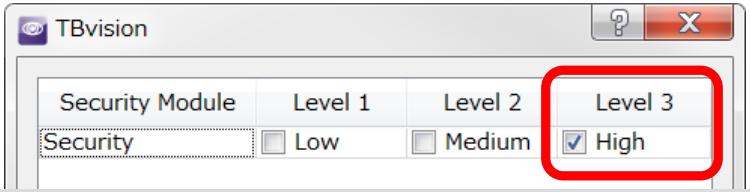
CERT 対応で他を圧倒！



Tool	No. of rules
A	7
B	96
C	122
D	50
E	11
F	63
G	3
H	48
I	21
J	135
LDRA tool suite	**201**
L	10
M	26

LDRAはシステムワイドにコントロールフローとデータフローを解析できることが優位性のひとつ

リスクベースのアプローチ



Security Qualification : demo_set

	Number Detected	Security Qualification Status
fio_get		
Global pointer not checked within this procedure. : filename1		High
fio_set		
File pointer not checked for null before use. : fwptr		High
Global pointer not checked within this procedure. : filename2		High
Use of sizeof on an array parameter.		High
Array index not unsigned.		High
demo2_main		
Array bound exceeded. : top_array[*]; accessed=27, range=0-26		High
Use of banned function, type or variable. : malloc		High
Pointer not checked for null before use. : ptr		High
demo3.c		
set_value		
initialise		
get_port		
update_value		
Procedure contains UR data flow anomalies.		High
Procedure contains UR data flow anomalies. : tolerance		High
Incorrect array index validation. : pv		High
finale		
Use of banned function, type or variable.	4	High
Pointer not checked for null before use.	2	High
free called on variable with no allocated space. : u16a_ptr		High
Free called twice on same variable. : sptr		High

Code Reviews

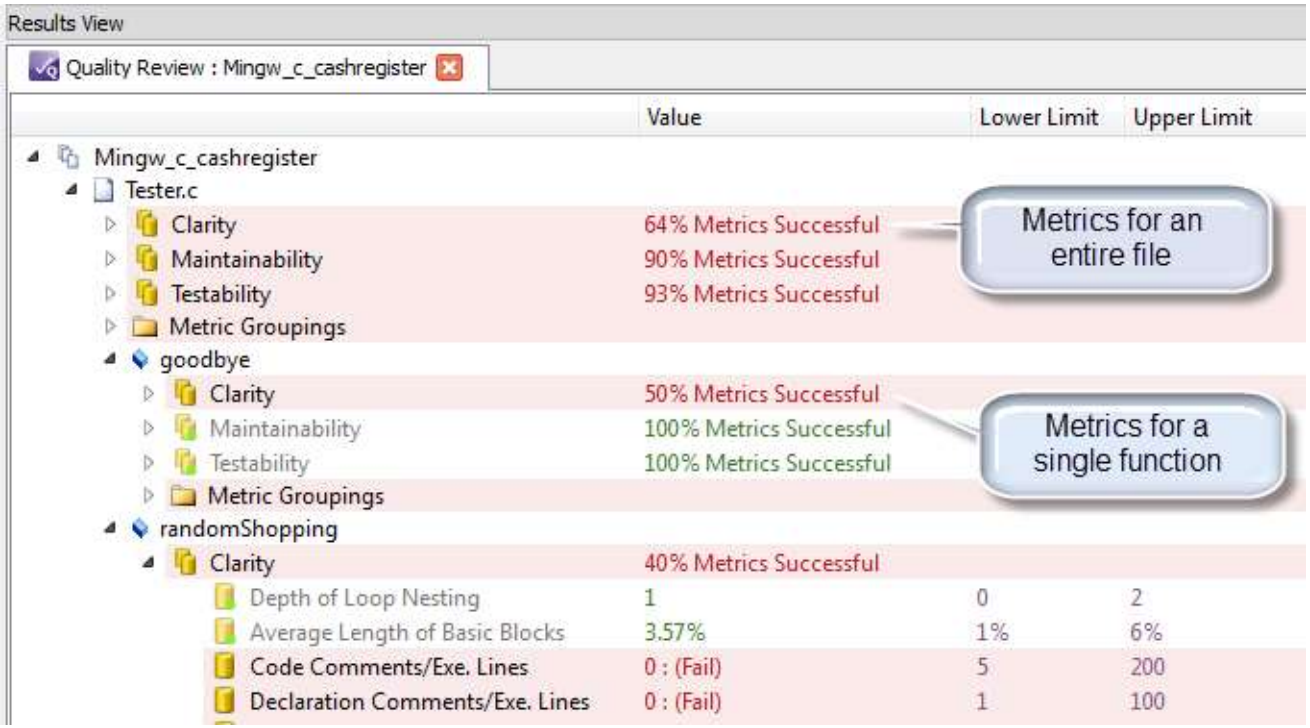
– Keep it simple

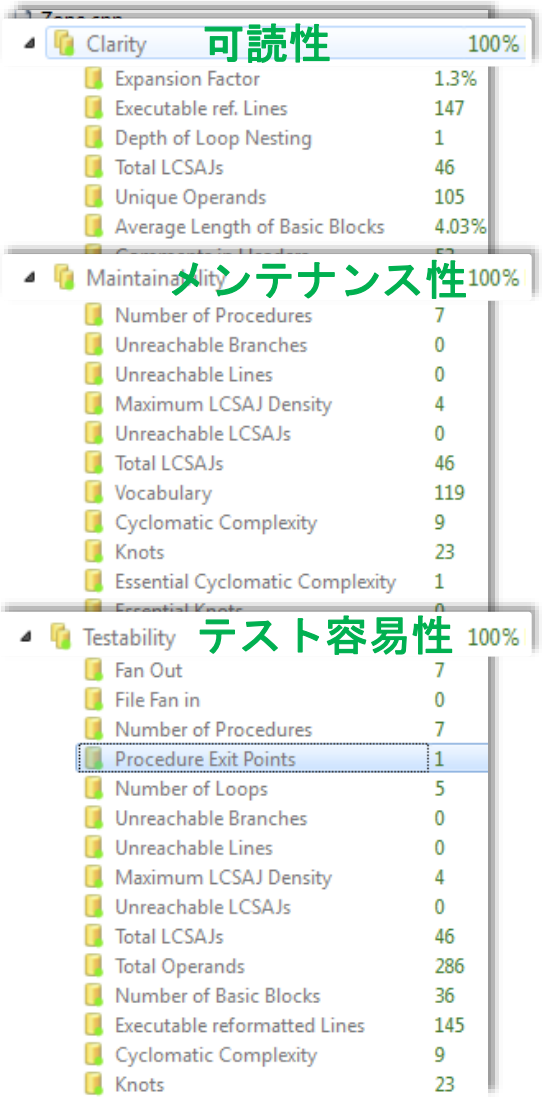
– Use effective quality assurance

複雑さがもたらす問題

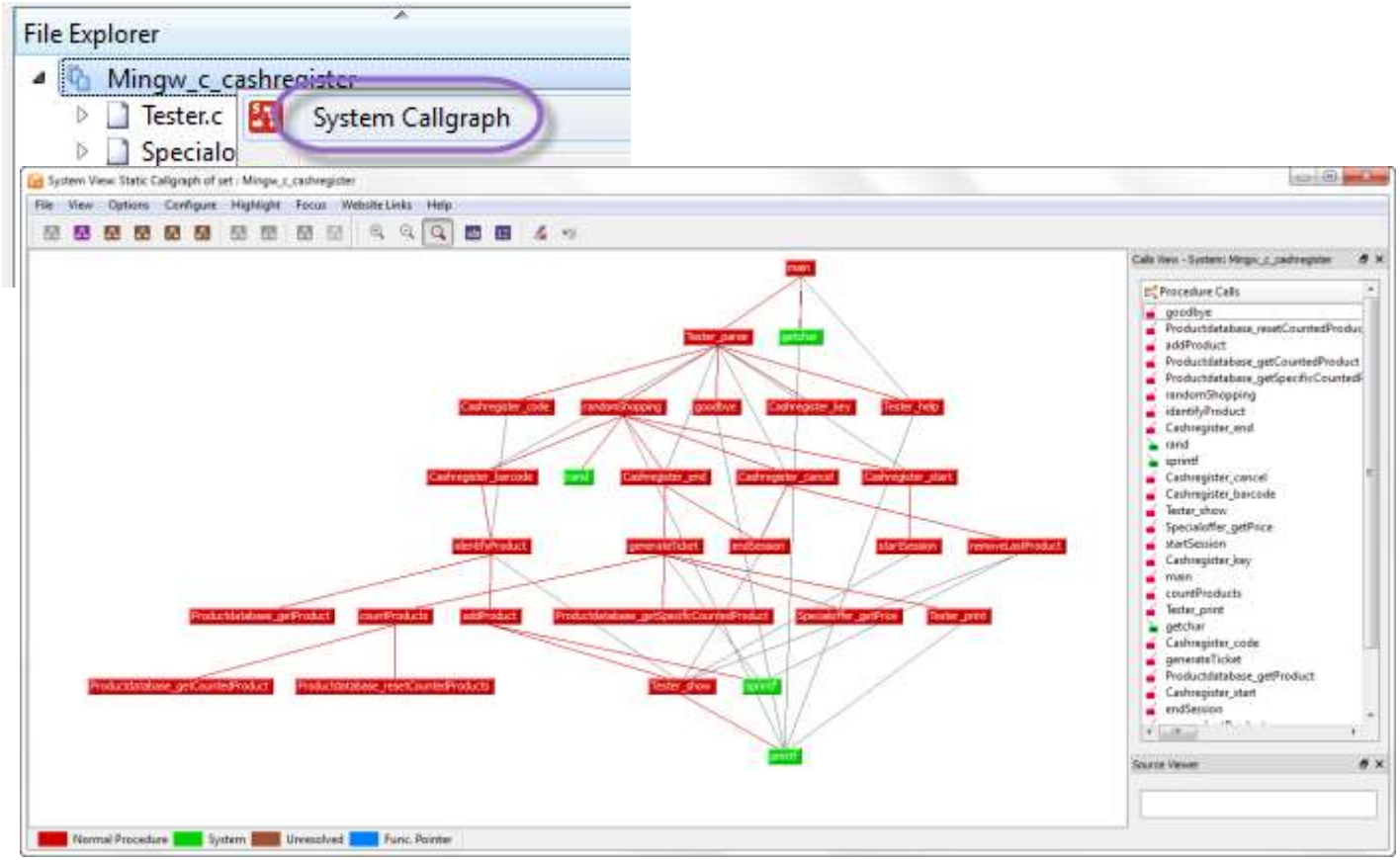
- 可読性が低下して脆弱性が奥に潜んでしまう
- テストは容易で無くなり余計な作業が必要
- 保守性が低く新たな脆弱性への迅速な対応が困難

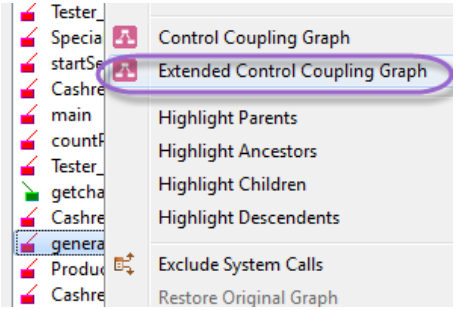
コード品質を自動解析

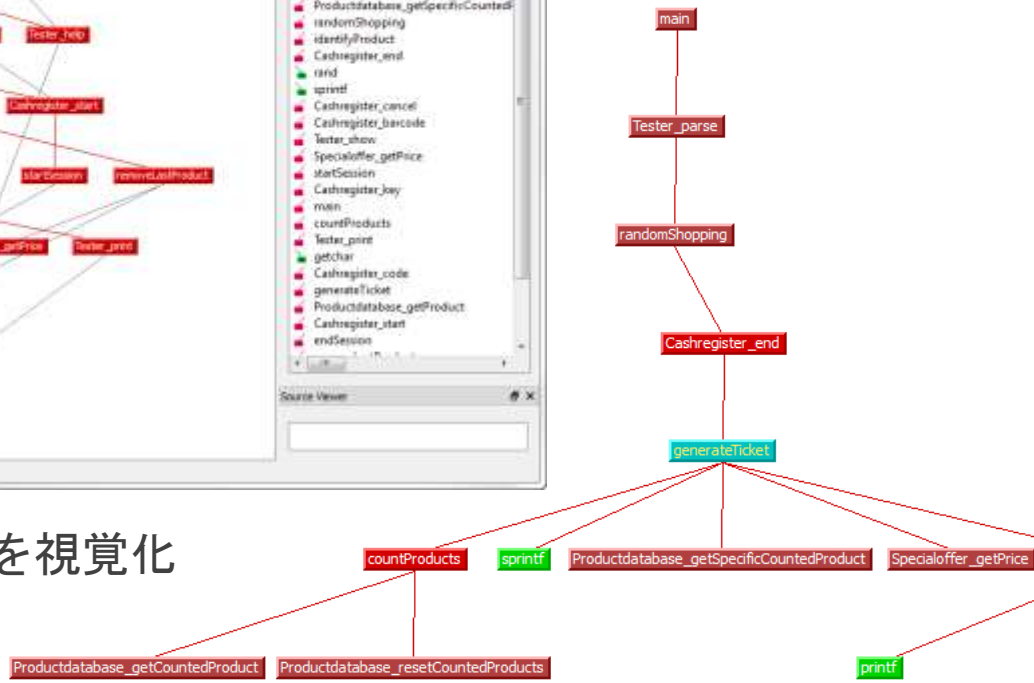

Results View
Quality Review : Mingw_c_cashregister
Value
Lower Limit
Upper Limit
Mingw_c_cashregister
Tester.c
Clarity
64% Metrics Successful
Maintainability
90% Metrics Successful
Testability
93% Metrics Successful
Metric Groupings
goodbye
Clarity
50% Metrics Successful
Maintainability
100% Metrics Successful
Testability
100% Metrics Successful
Metric Groupings
randomShopping
Clarity
40% Metrics Successful
Depth of Loop Nesting
1
0
2
Average Length of Basic Blocks
3.57%
1%
6%
Code Comments/Exe. Lines
0 : (Fail)
5
200
Declaration Comments/Exe. Lines
0 : (Fail)
1
100
Metrics for an entire file
Metrics for a single function


Clarity
可読性
100%
Expansion Factor
1.3%
Executable ref. Lines
147
Depth of Loop Nesting
1
Total LCSAJs
46
Unique Operands
105
Average Length of Basic Blocks
4.03%
Maintainability
メンテナンス性
100%
Number of Procedures
7
Unreachable Branches
0
Unreachable Lines
0
Maximum LCSAJ Density
4
Unreachable LCSAJs
0
Total LCSAJs
46
Vocabulary
119
Cyclomatic Complexity
9
Knots
23
Essential Cyclomatic Complexity
1
Testability
テスト容易性
100%
Fan Out
7
File Fan in
0
Number of Procedures
7
Procedure Exit Points
1
Number of Loops
5
Unreachable Branches
0
Unreachable Lines
0
Maximum LCSAJ Density
4
Unreachable LCSAJs
0
Total LCSAJs
46
Total Operands
286
Number of Basic Blocks
36
Executable reformatted Lines
145
Cyclomatic Complexity
9
Knots
23

コントロールカップリング





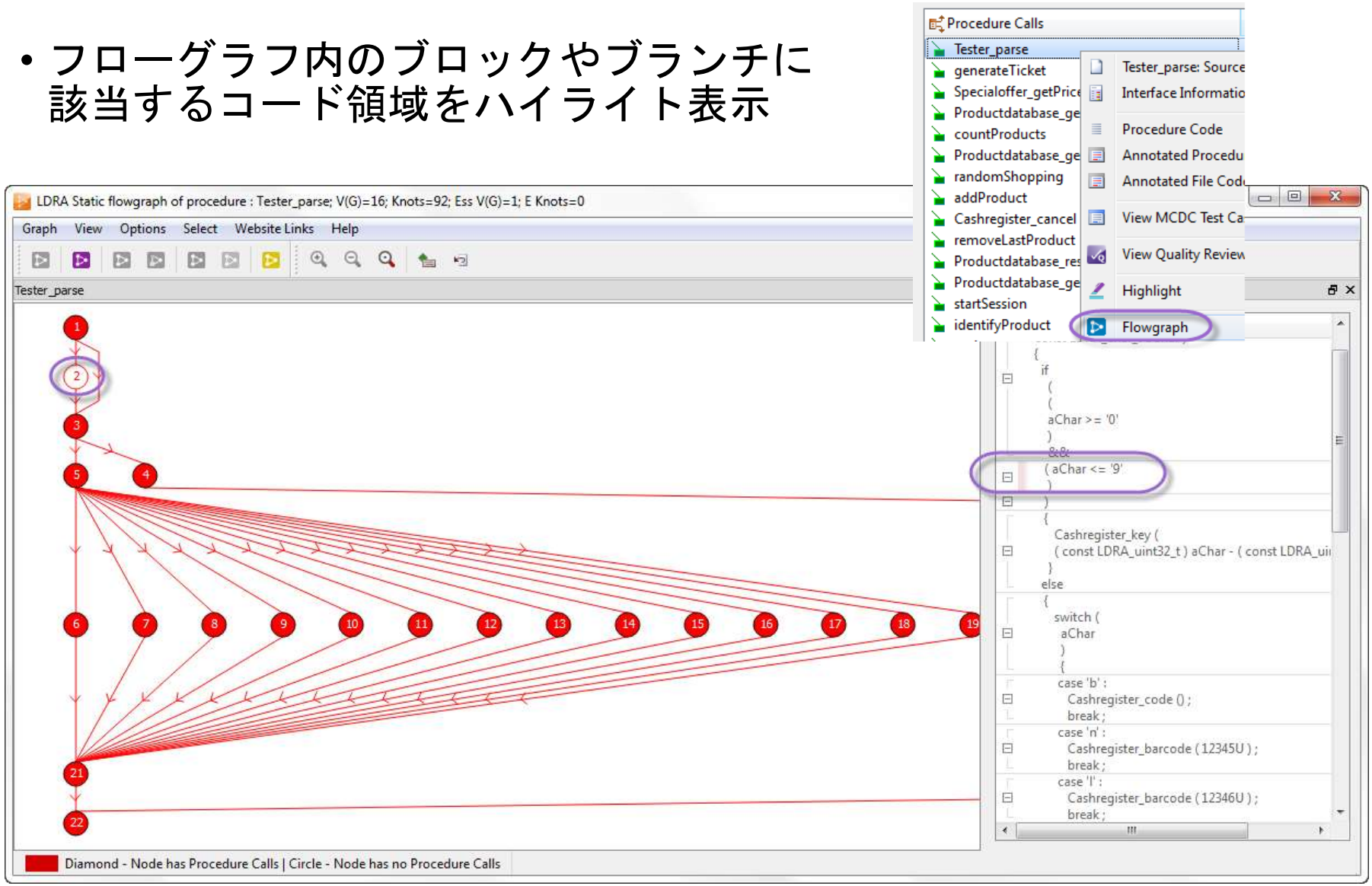


System Callgraph 関数間の相互関係を視覚化

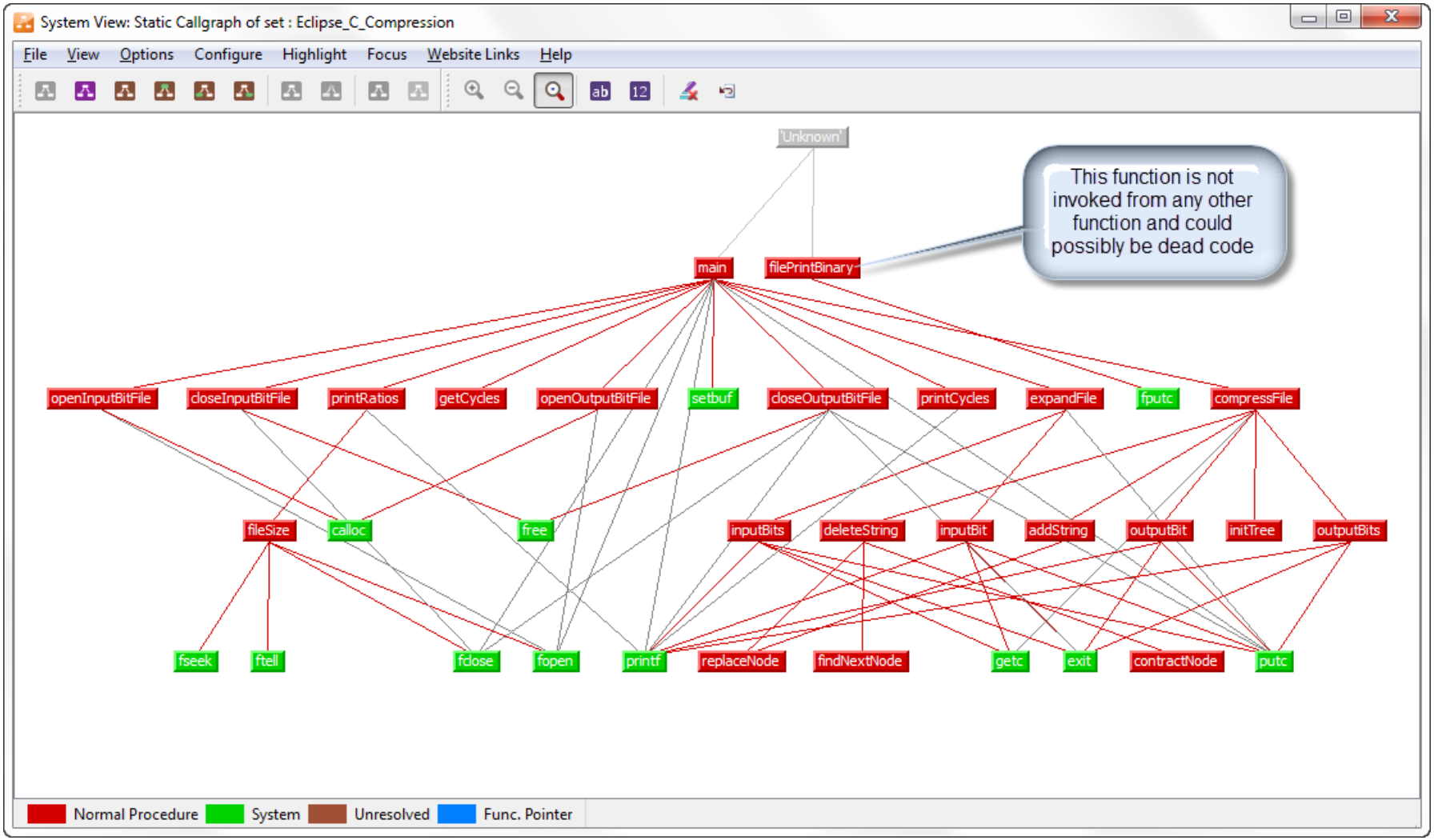
Extended Control Coupling Graph 特定関数にフォーカス

フローグラフ

- フローグラフ内のブロックやブランチに該当するコード領域をハイライト表示



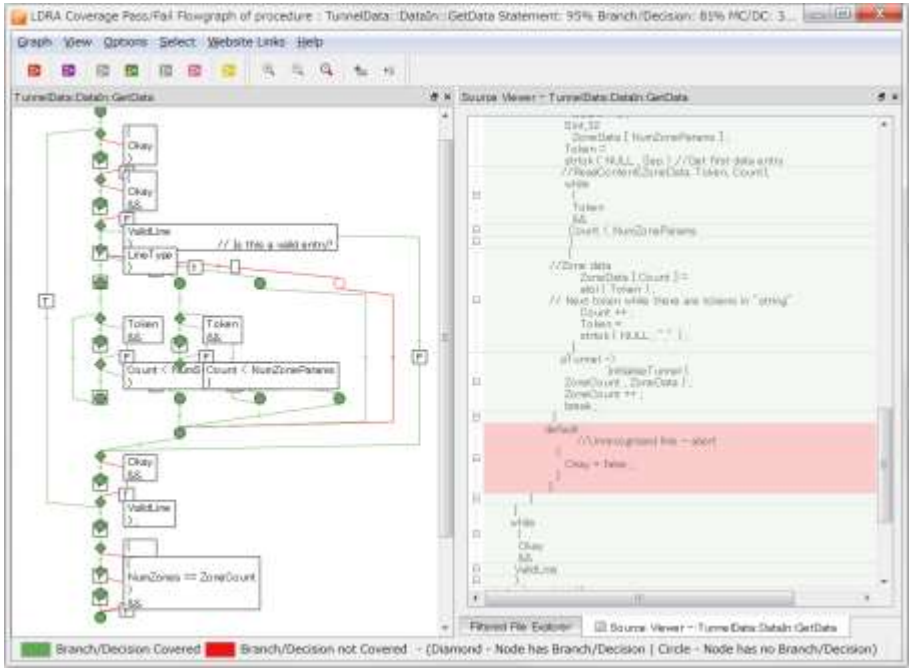
Static Callgraph – デッドコードの検出



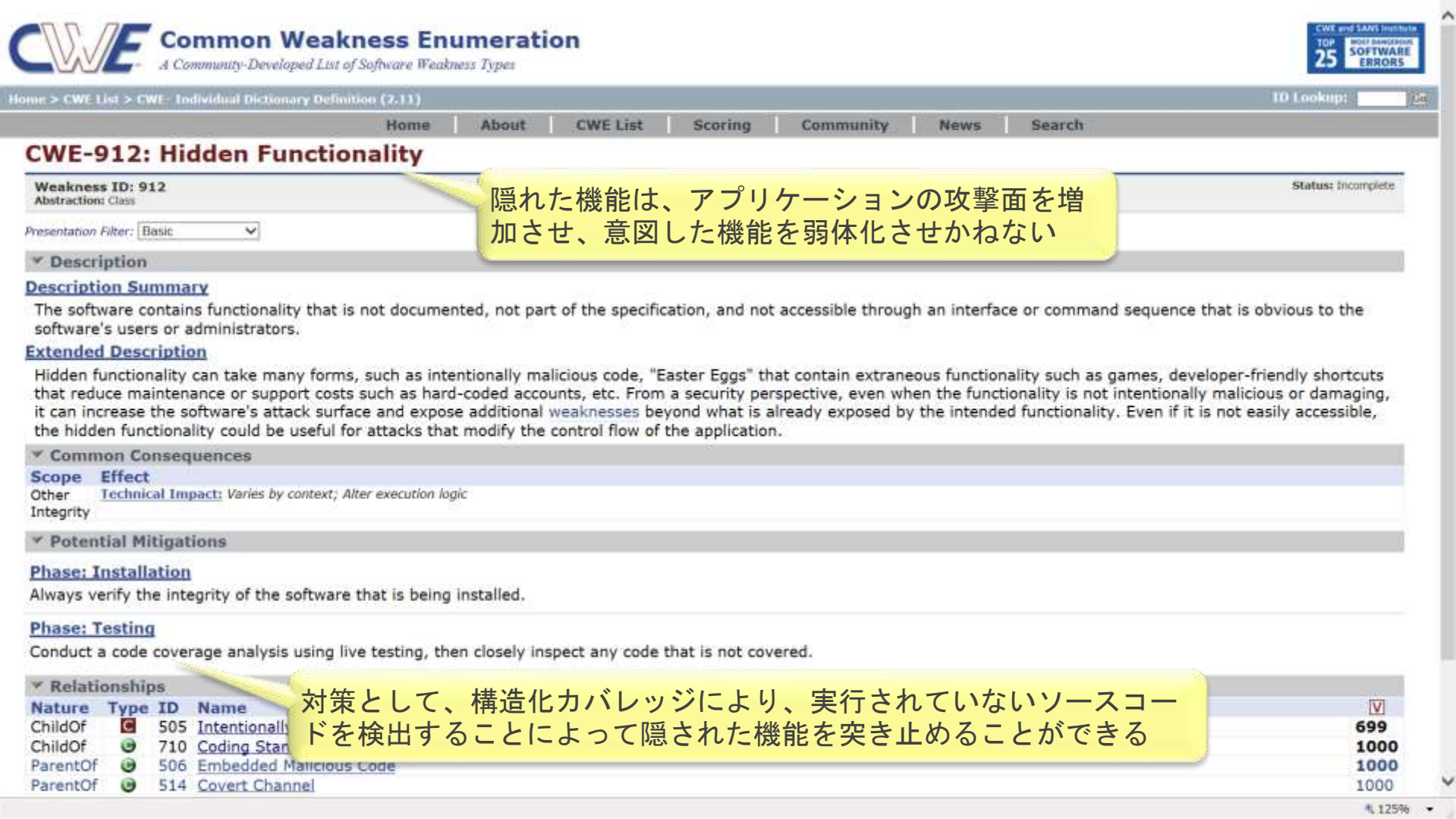
Test Coverage

テストのカバレッジ

- セキュアコーディングの実践により「防御的」ソフトウエアを得られたとして、それが動作することをテストする必要がある
- そのテストの十分性は？
- 安全性レベルに応じた各種カバレッジレベルに対応
  - Entry points coverage
  - Statement coverage
  - Branch decision coverage
  - Modified Condition / Decision Coverage (MC/DC)
  - Data coupling and control coupling coverage
  - Object code coverage
  - Linear Code Sequence And Jump coverage (LCSAJ)



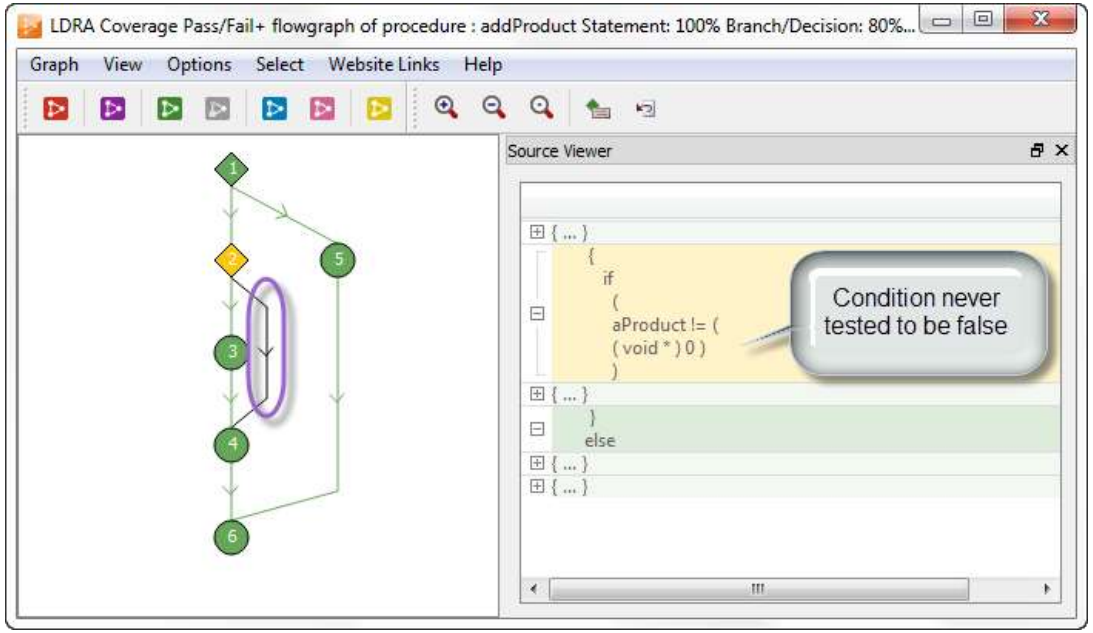
CWE-912 隠れた機能



https://cwe.mitre.org/data/definitions/912.html
http://jvndb.jvn.jp/ja/contents/2015/JVNDB-2015-006032.html

100%テストカバレッジの重要性

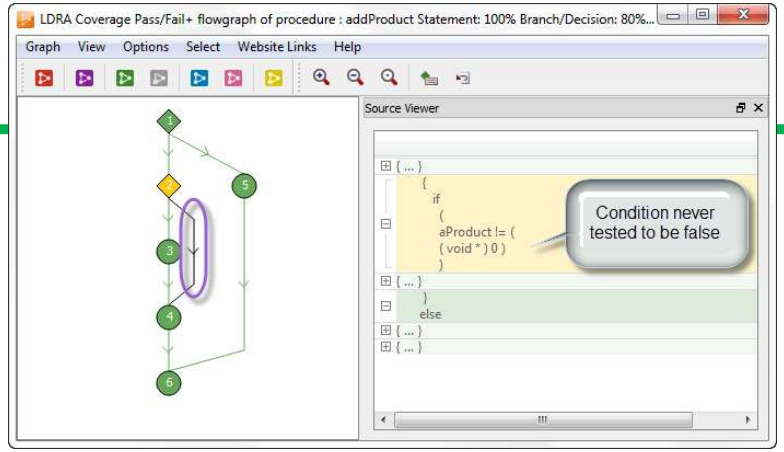
- 一般に、例外的な処理をするコードは十分にテストされていない可能性が高く、めったにない条件下で、安全性やセキュリティの問題を引き起こすことになる
- 例えばポインタがNULLでない場合など、ディフェンシブなプログラムはシステムテストでは確認できない
- この例で100%カバレッジを満たすには単体テストによる評価が必要



Importance of Structural Coverage

システムレベルテストで満たされないパスを

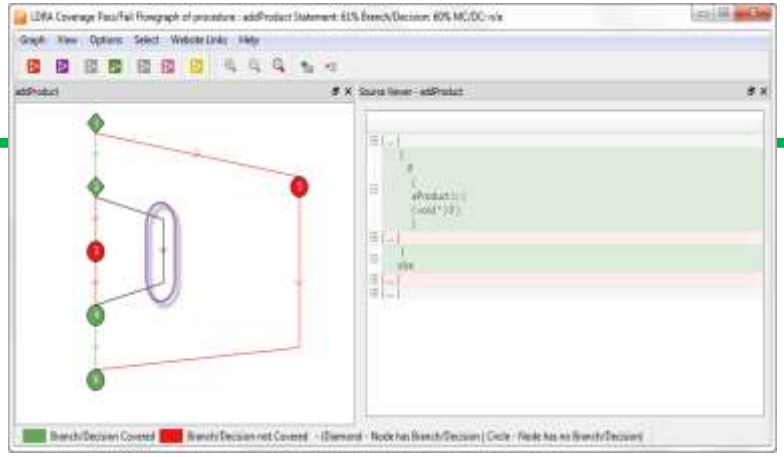
HLR_0001
HLR_0002
…..
HLR_000n



HLT_0001
HLT_0001
…..
HLT_000m

単体テストで満たす

LLR_0001
LLR_0002
…..
LLR_000n

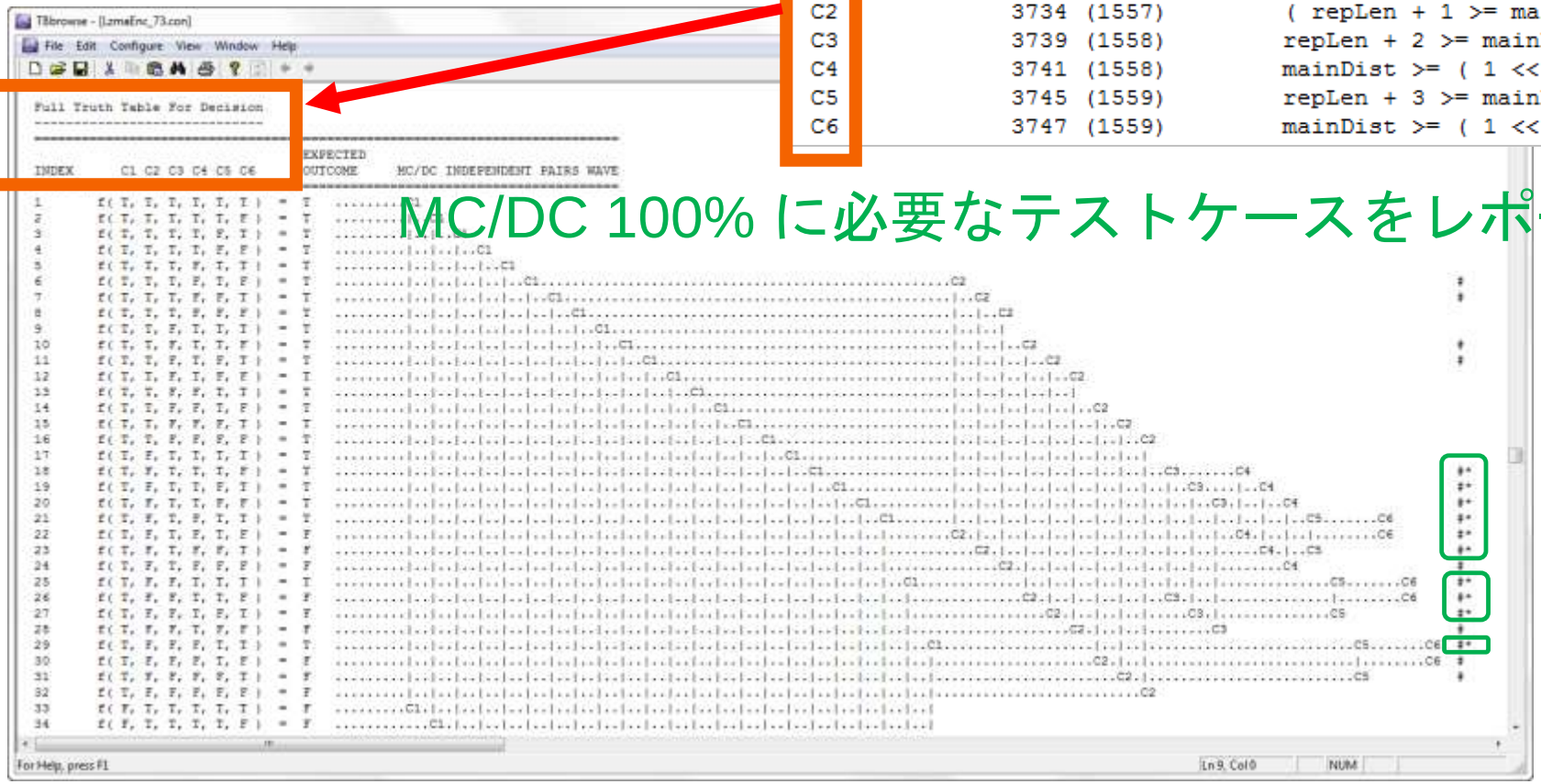


LLT_0001
LLT_0001
…..
LLT_000m

MC/DC テストケース解析

- Table B.2 7d - Structural test coverage (conditions, MC/DC) 100 % - Partial - 2 artifacts
  - Code coverage report fulfilled by 1 item
    - Mingw_c_cashregister.frm.htm
  - MC/DC test plan report fulfilled by 1 item
    - MCDC_Planner_Report.txt

```
Branch Condition List
---------------------
=========================================
CONDITION      LINE NUMBER        CONDITION
NUMBER         REF. (SOURCE)      TEXT
=========================================
C1             3731 (1556)        repLen >= 2
C2             3734 (1557)        ( repLen + 1 >= mainLen )
C3             3739 (1558)        repLen + 2 >= mainLen
C4             3741 (1558)        mainDist >= ( 1 << 9 )
C5             3745 (1559)        repLen + 3 >= mainLen
C6             3747 (1559)        mainDist >= ( 1 << 15 )
```



MC/DC 100% に必要なテストケースをレポート

Test for Security Requirements

要件トレーサビリティ

すべての機能、安全およびセキュリティの要件が
正しく実装され検証されることを保証するメカニズム

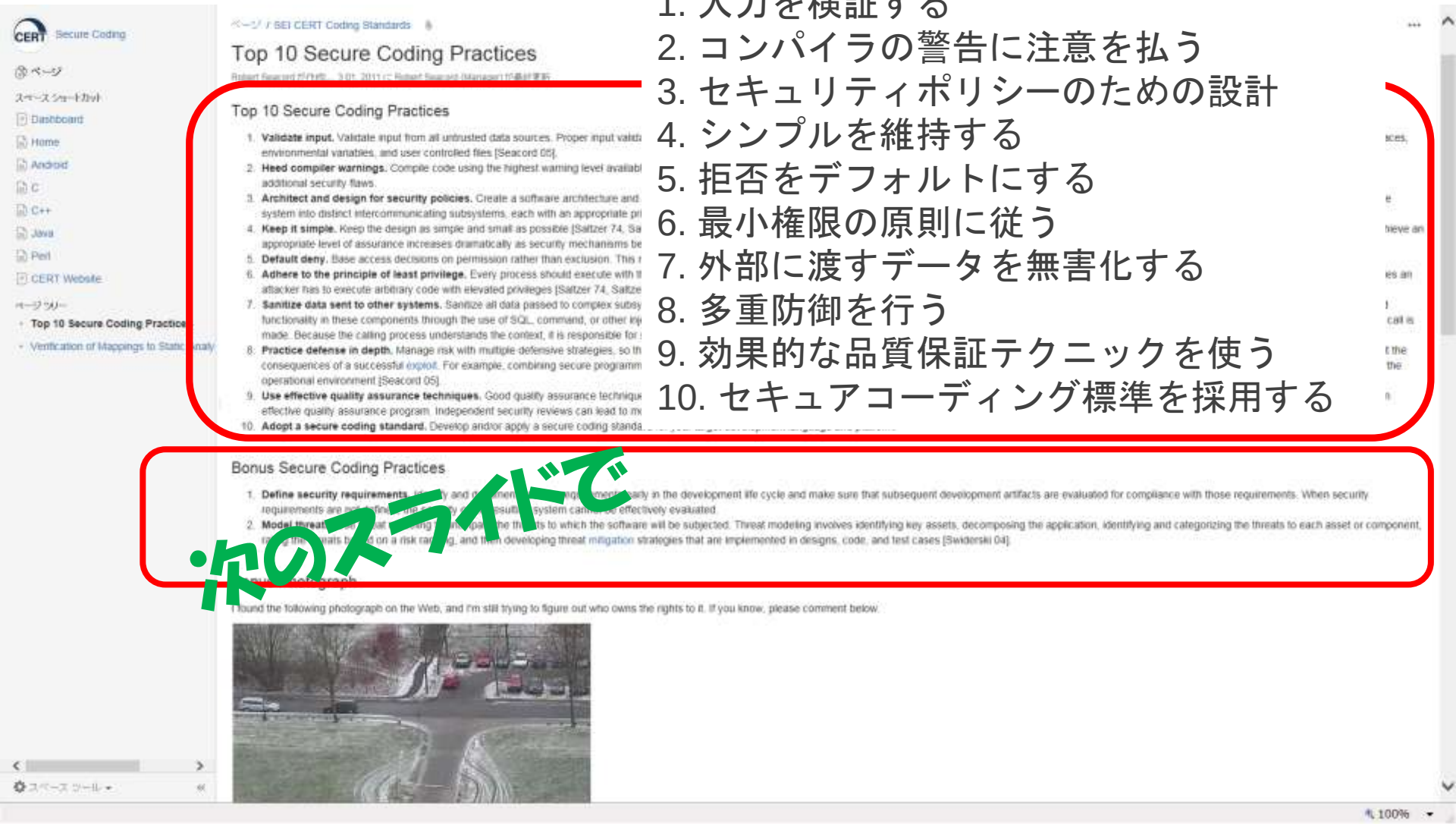
Security Requirements
Safety Requirements

要件にはないコードの存在が欠陥や脆弱性の原因になる
ことは多い。利用されないなら削除されるべき

双方向のトレーサビリティにより、すべての要件を満た
すためのコードが存在することが保証される

CERT's Top 10 Secure Coding Practices

1. 入力を検証する
2. コンパイラの警告に注意を払う
3. セキュリティポリシーのための設計
4. シンプルを維持する
5. 拒否をデフォルトにする
6. 最小権限の原則に従う
7. 外部に渡すデータを無害化する
8. 多重防御を行う
9. 効果的な品質保証テクニックを使う
10. セキュアコーディング標準を採用する



https://www.securecoding.cert.org/confluence/display/seccode/Top+10+Secure+Coding+Practices

Bonus Secure Coding Practices

セキュリティ要求を定義する

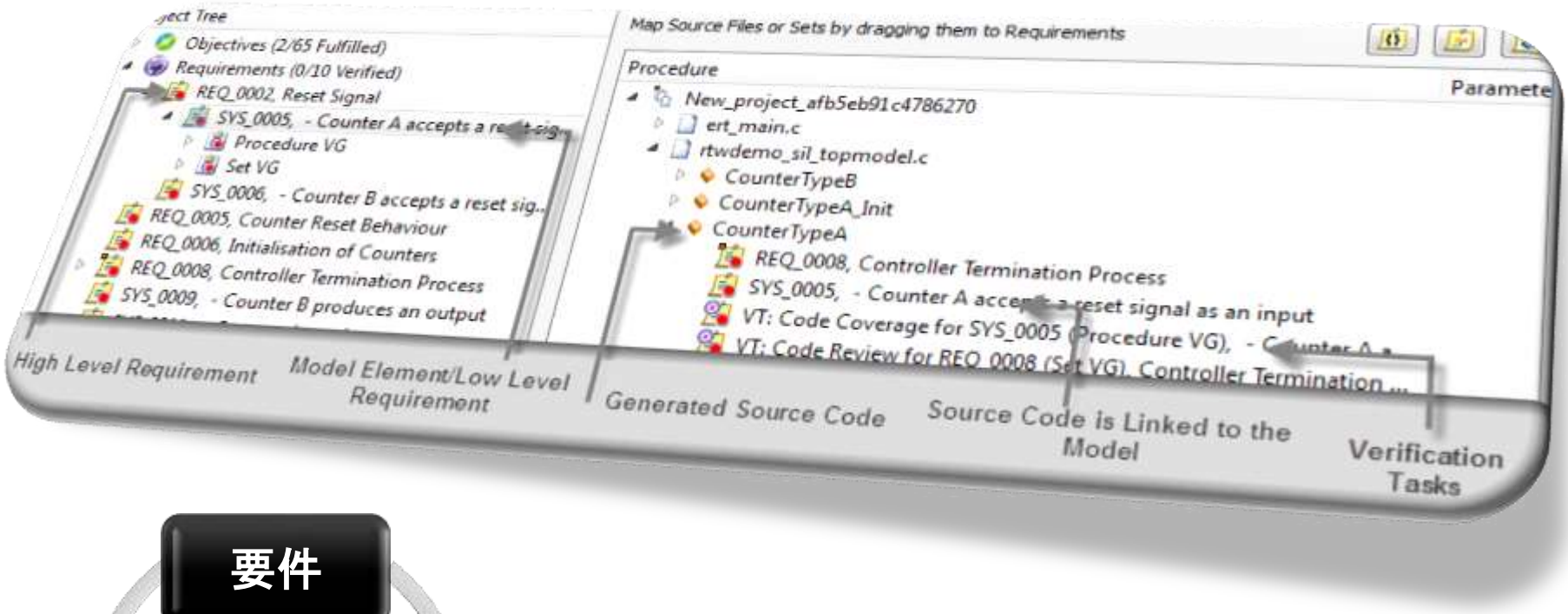
- 開発ライフサイクルの早期段階でセキュリティ要件を特定し文書化して、後続の開発成果物がそれら要件に従っていることを評価する。 セキュリティ要件が定義されない場合、そのシステムのセキュリティを効果的に評価することはできない。

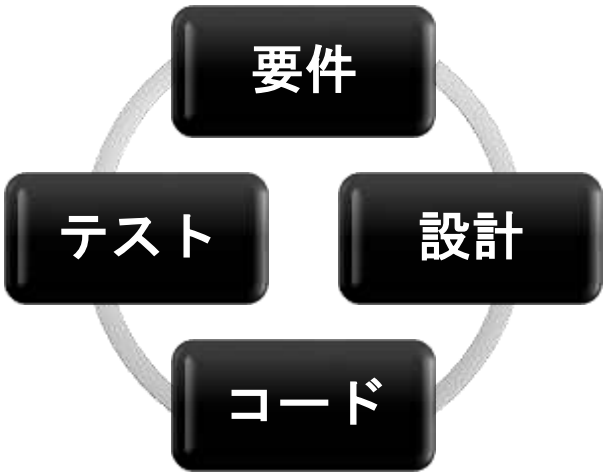
脅威をモデル化する

- 脅威をモデリングすることで、ソフトウェアが受ける脅威を予測する。 脅威のモデリングでは、主要な資産の特定、アプリケーションの分類、各資産またはコンポーネントへの脅威の特定と分類、脅威をリスクレベルで格付け、そして脅威を緩和するための戦略を、設計・コード・テストケースで実装する。

https://www.securecoding.cert.org/confluence/display/seccode/Top+10+Secure+Coding+Practices

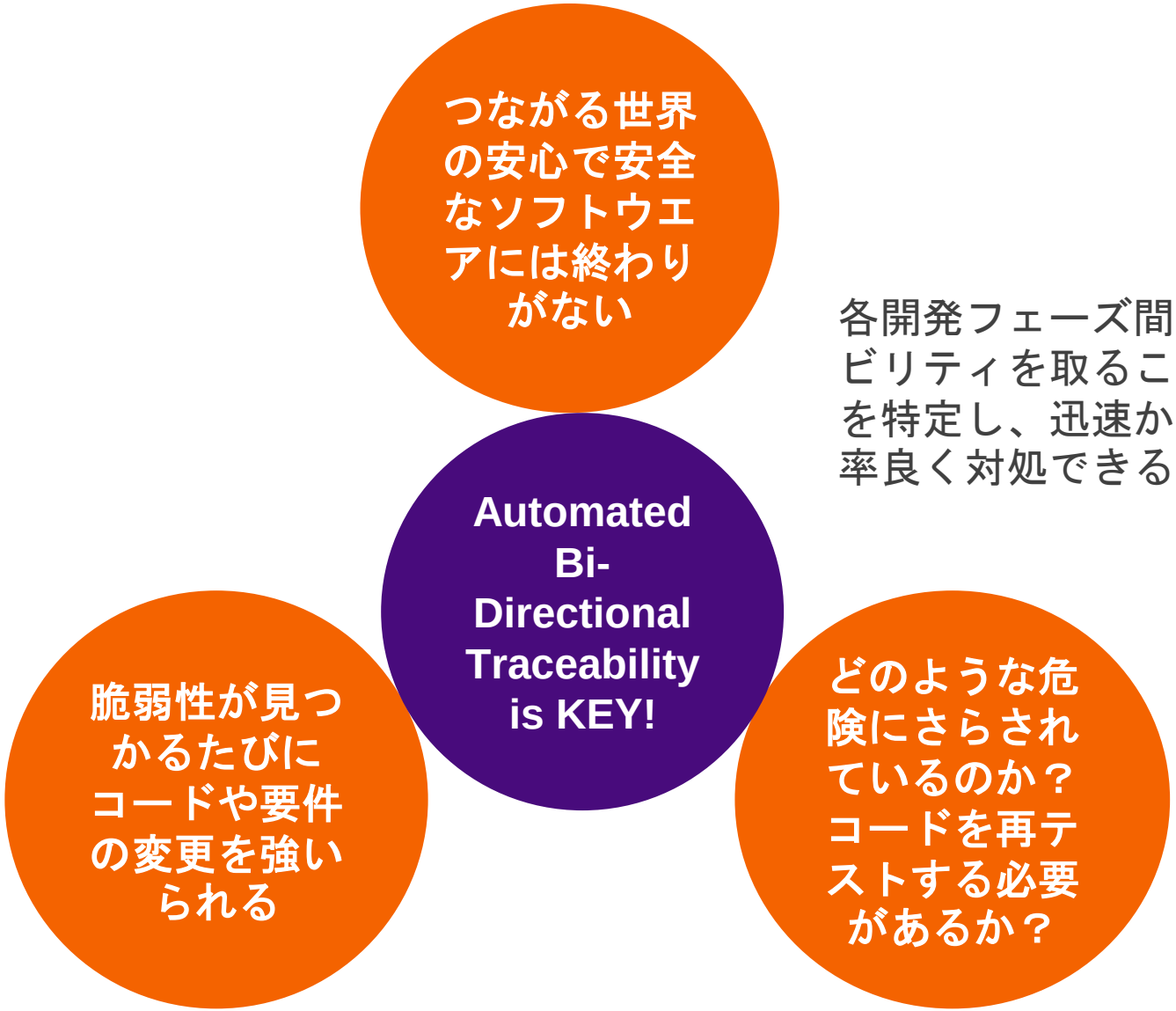
要件トレーサビリティ


Objectives (2/65 Fulfilled)
Requirements (0/10 Verified)
REQ_0002, Reset Signal
SYS_0005, - Counter A accepts a reset sig...
Procedure VG
Set VG
SYS_0006, - Counter B accepts a reset sig...
REQ_0005, Counter Reset Behaviour
REQ_0006, Initialisation of Counters
REQ_0008, Controller Termination Process
SYS_0009, - Counter B produces an output
Map Source Files or Sets by dragging them to Requirements
Procedure
New_project_afb5eb91c4786270
ert_main.c
rtwdemo_sil_topmodel.c
CounterTypeB
CounterTypeA_Init
CounterTypeA
REQ_0008, Controller Termination Process
SYS_0005, - Counter A accepts a reset signal as an input
VT: Code Coverage for SYS_0005 (Procedure VG), - Counter A a...
VT: Code Review for REQ_0008 (Set VG), Controller Termination ...
High Level Requirement
Model Element/Low Level Requirement
Generated Source Code
Source Code is Linked to the Model
Verification Tasks


要件
設計
コード
テスト

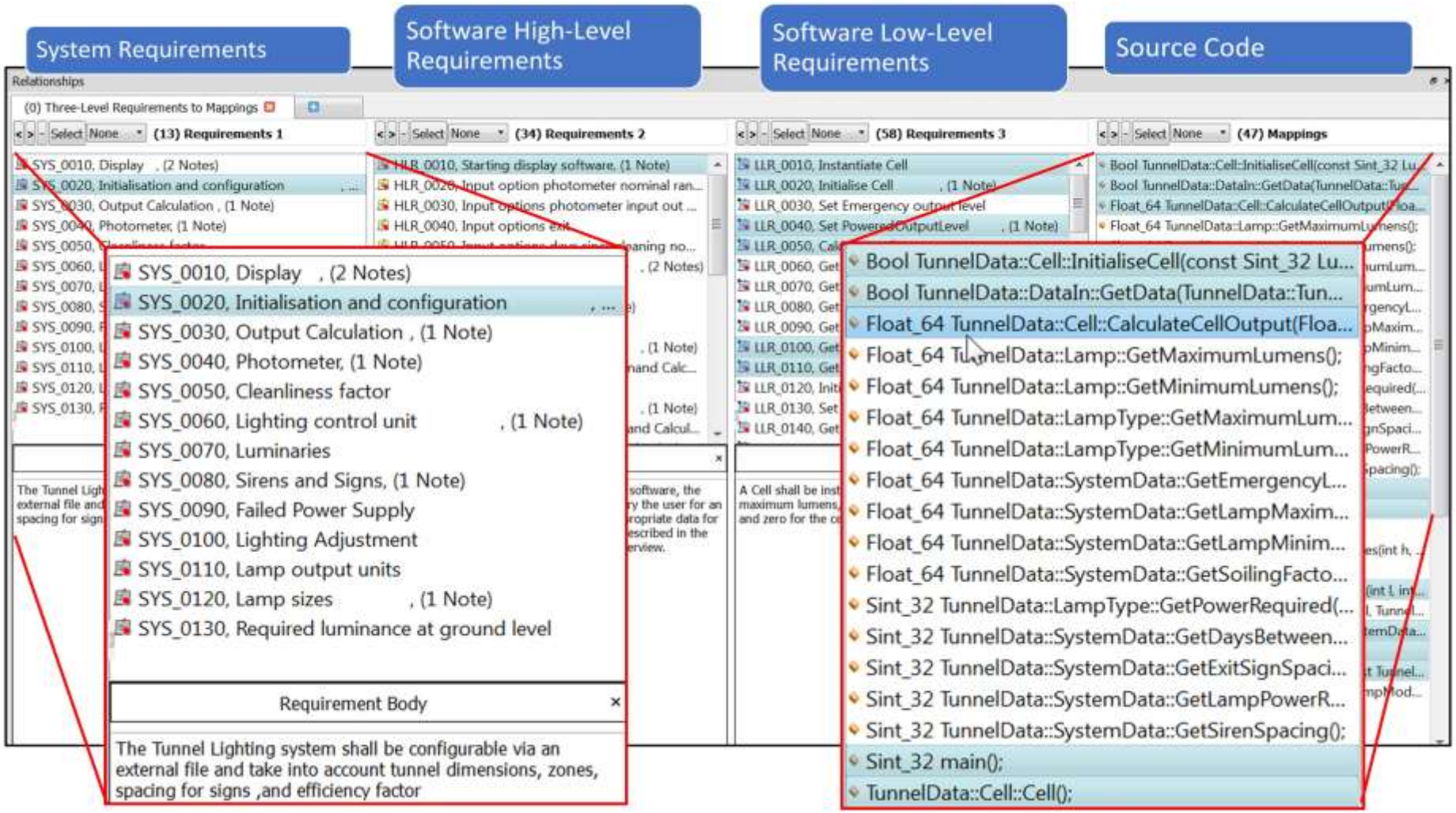
Maintenance

Secure Application Maintenance Phase



各開発フェーズ間のトレーサビリティを取ることで、問題を特定し、迅速かつコスト効率良く対処できるようになる

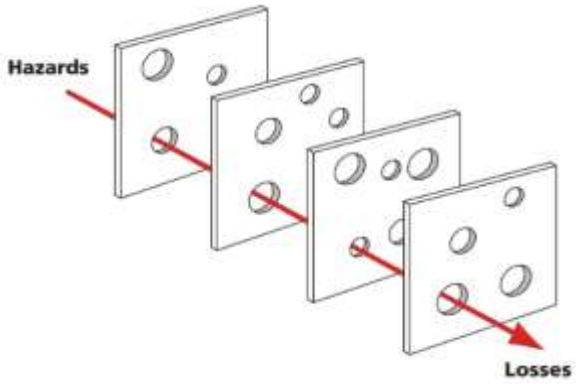
Automated Bi-Directional Traceability



要件～コード、テストまで変更のインパクトを成果物間で追跡

まとめ

セキュリティを組み込む



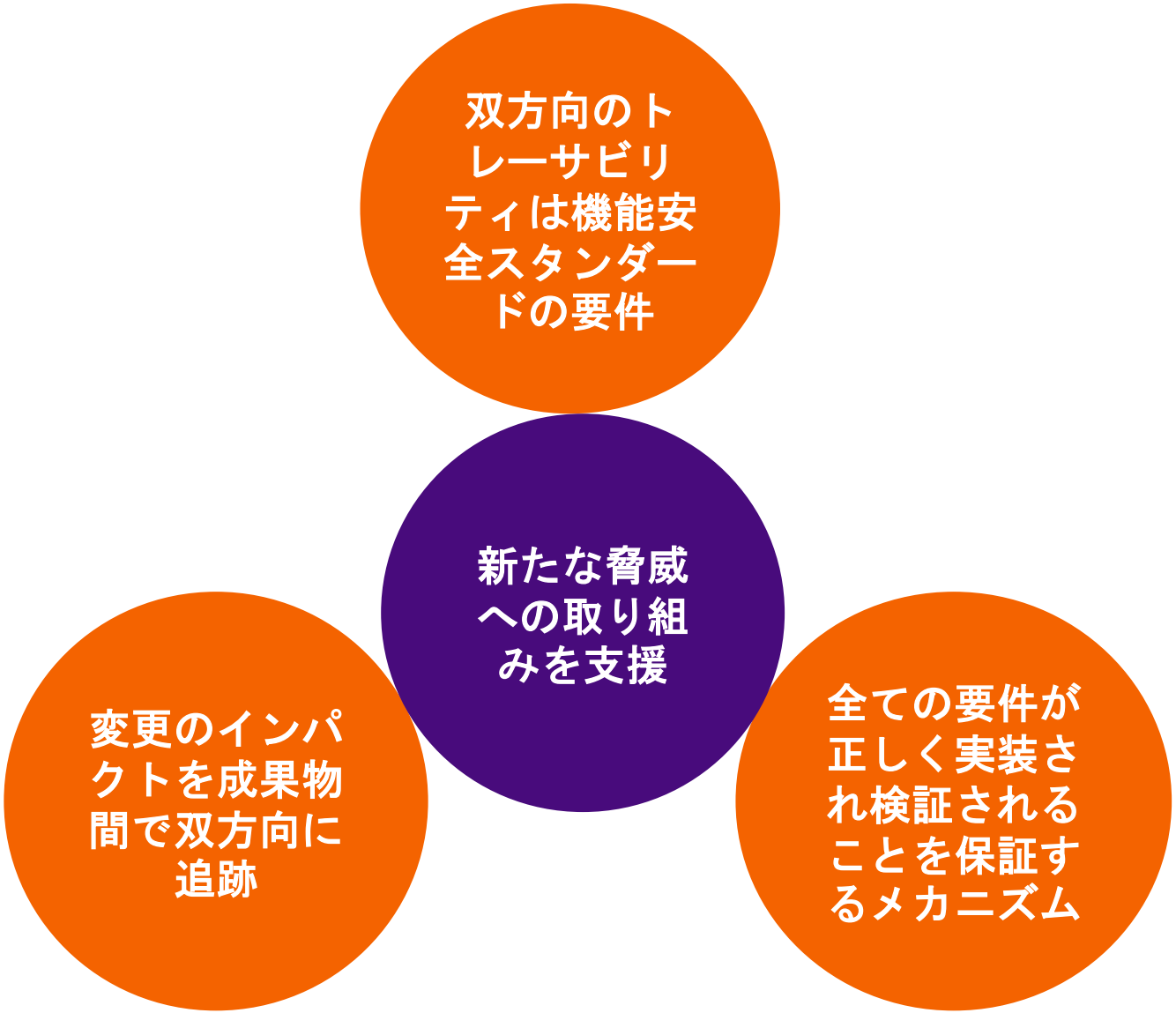
セキュリティ対策に単一解は無い

セキュアなコードはスイスチーズモデルの防御に不可欠

防御技術にかかわらず、アプリケーションはシステムの重要な役割を果たす

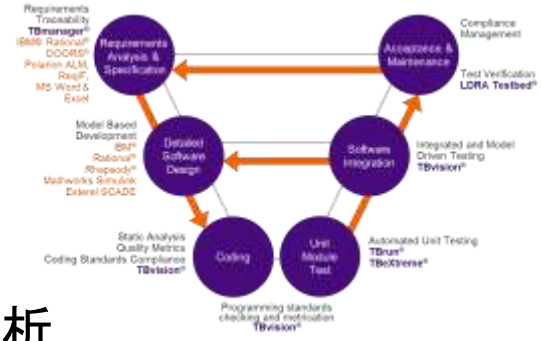
つながる世界の安心で安全なソフトウエアには、多重防御が必要

トレーサビリティ


双方向のトレーサビリティは機能安全スタンダードの要件
新たな脅威への取り組みを支援
変更のインパクトを成果物間で双方向に追跡
全ての要件が正しく実装され検証されることを保証するメカニズム

各機能が有機的につながる

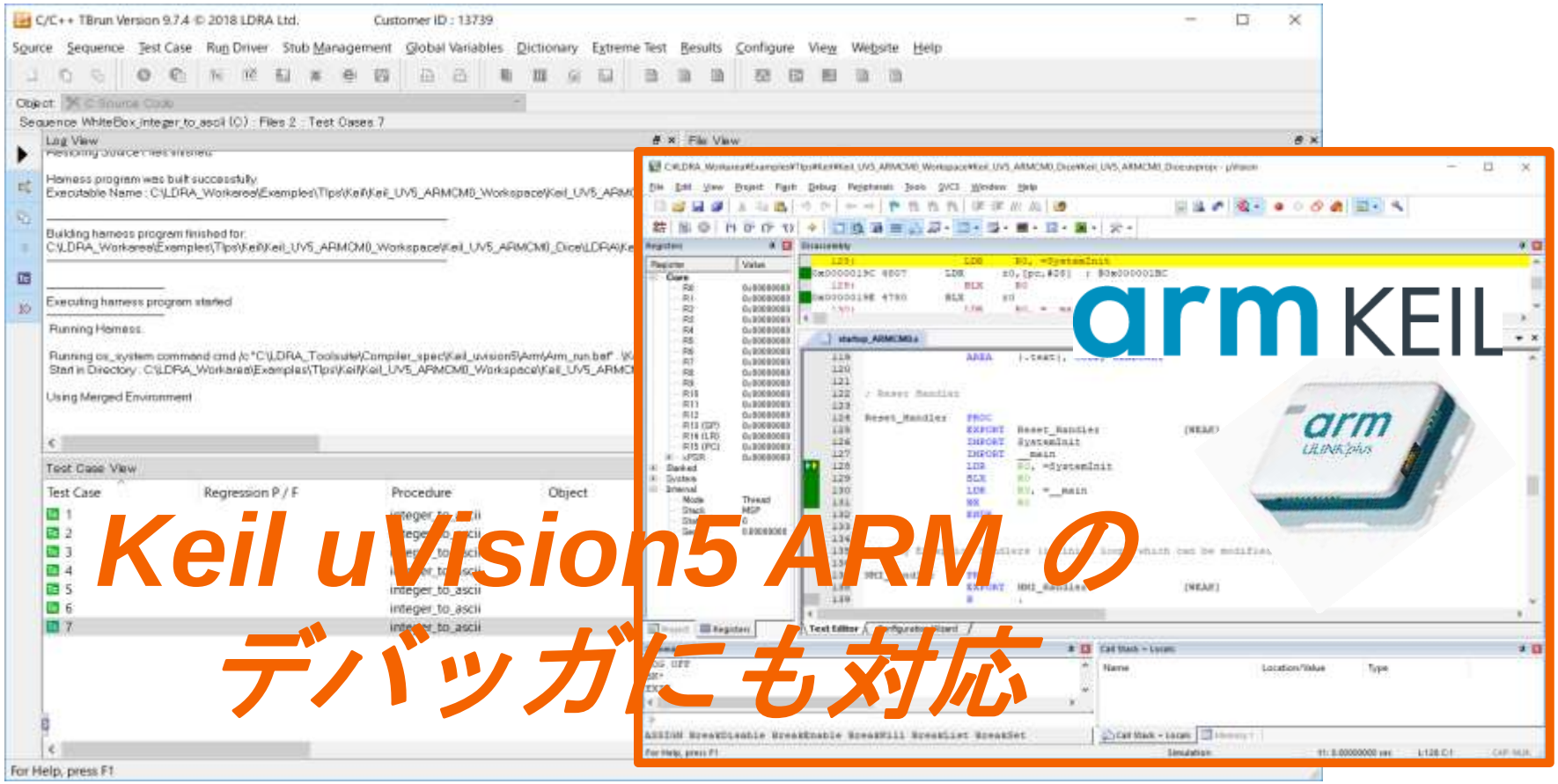
- 静的解析：コーディングスタンダード準拠、コード品質のメトリクス
  - MISRA 、CERT C、IPA/SEC ESCR 等各種サポート。ユーザ定義ルール集も編集できる
  - コードの視覚化（コールグラフ、フローグラフ、クロスリファレンス）
- 動的解析：構造化カバレッジ
  - 関数コール、ステートメント、ブランチ、MC/DC 、LCSAJ (path coverage).
  - アセンブラレベルカバレッジ (DO-178C Level A)
  - あらゆる組込みコンパイラ、実行環境をサポート
- 単体テスト（Unit/low-level testing）
  - テストハーネス、スタブ、グローバル宣言などの自動生成
  - 堅牢性テストのためのテストケース自動生成



- データフロー・コントロールフローの静的・動的解析
  - DO-178C で認定可能な唯一の汎用ツール
- 要件トレーサビリティに検証実行・オブジェクティブ管理を統合
- ツール認定キットを提供（DO-178Cに必要）
- TUVの事前認定取得済み

https://www.fuji-setsu.co.jp/products/LDRA/

あらゆるターゲット環境に対応



Keil uVision5 ARM 環境で静的解析～動的テスト実行とカバレッジ解析
https://www.fuji-setsu.co.jp/demo/LDRAKeiluVision5ARM.wmv

Keil uVision5 ARM 環境で要件トレーサビリティとテスト実行
https://www.fuji-setsu.co.jp/demo/TBmanager_KeiluVision5_ARM.wmv

スタンダード認証取得を目的に広く採用

LDRA

LDRA 社は各種国際スタンダードに貢献

Professor Mike Hennell

Member of SC-205 / WG-71 (DO-178C) formal methods subgroup

Member of MISRA C committee and MISRA C++ committee

Member of the working group drafting a proposed secureC annex for the C language definition (SC 22 / WG14)

Bill St Clair

Member of SC-205 / WG-71 (DO-178C) Object Oriented Technology subgroup

Andrew Banks

MISRA Committee Chair

Committee Member for Second Edition of ISO 26262

Chair of BSI IST/15/-/26 for Software Testing

MISRA representative to BSI IST/15 for Software and Systems Engineering

Member of BSI IST/15-/20 for Bodies of Knowledge and Professionalization

LDRA 社は各種国際スタンダードに貢献

Dr Clive Pygott

Member of ISO software vulnerabilities working group (SC 22 / WG 23)

Member of MISRA C++ committee

Member of the working group drafting a proposed secureC annex for the C language definition (SC 22 / WG14)

Liz Whiting

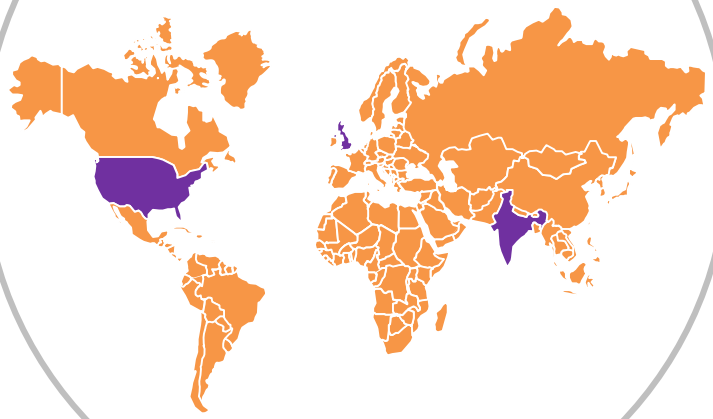
Member of MISRA C committee language definition

Chris Tapp

Chairman of MISRA C++ committee

Member of MISRA C committee language definition

LDRA



ISO 9001認証取得

1975年設立 => 40年以上の実績とブランド

IEC 61508, IEC 62304, EN 50128,

ISO 26262, IEC 60880 のツール認定取得

スタンダード認証プロセスを支援する

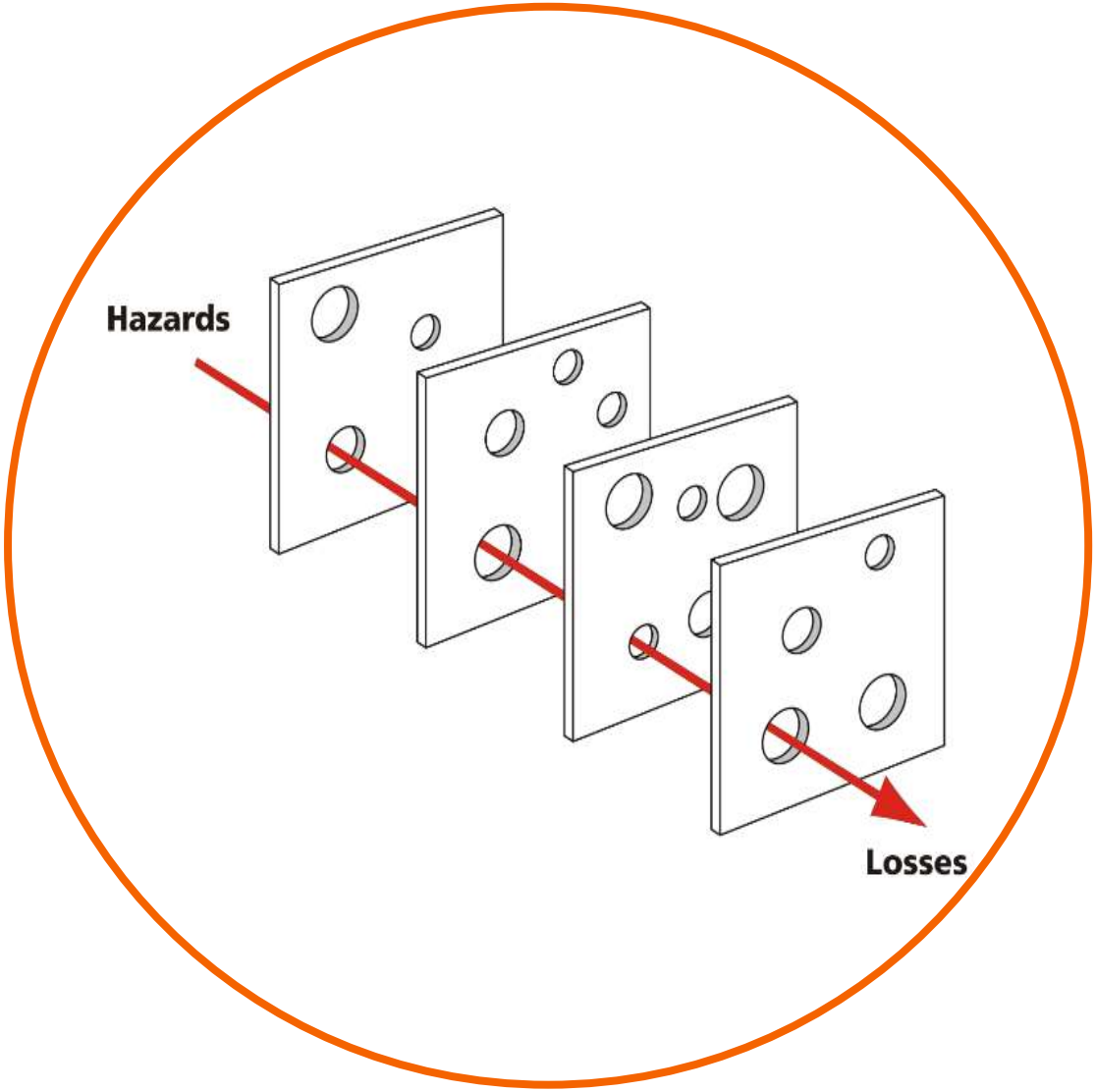
ソフトウエアテスト、管理支援ツールを提供

各種スタンダード委員会に貢献

e.g. DO-178C,

MISRA C/C++, CERT

コンパイラのテスト



C/C++コンパイラをテストして妥当性を検証する

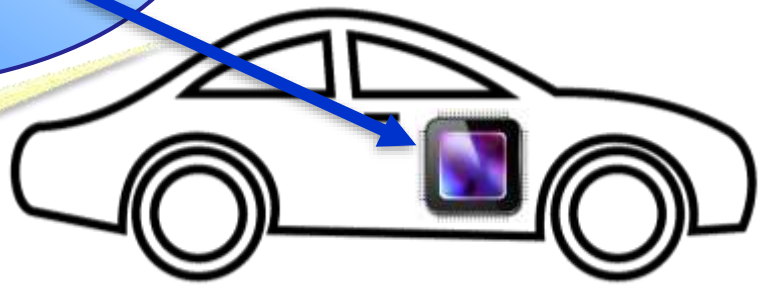
～ ユーザーまでもがコンパイラをテストすべき理由とは？ ～

コンパイラテストの重要性

コンパイラは巨大で非常に複雑なソフトウエア
- 品質の検証には膨大な条件の組合わせによるテストが必要
- コンパイラメーカは全オプションの組合せをテストできない

ブレーキに
対する
Cソースコード

C/C++コンパイラ
500万行規模



間違ったコードが生成されることによる被害が甚大

SuperTest™

- C，C++言語標準規格への適合性、正確性、堅牢性をチェックする300万件以上のテスト
- 30年以上の経験の積み重ねで開発
- 主にポジティブとネガティブテスト

・・・

コード生成機能を評価するテスト

診断機能を評価するテスト

コンパイラユーザも採用

顧客事例　https://www.fuji-setsu.co.jp/products/SuperTest/#customers

コンパイラユーザ成功例

- 同じコンパイラを1000本以上使用している
  - バージョンアップごとにSuperTestでチェックして、使ってはいけない最適化などの社内ルールを施行することで、過去のバージョンに戻さなければならないような問題が起こっていない。数千人に関わることなので非常に高い成果である。

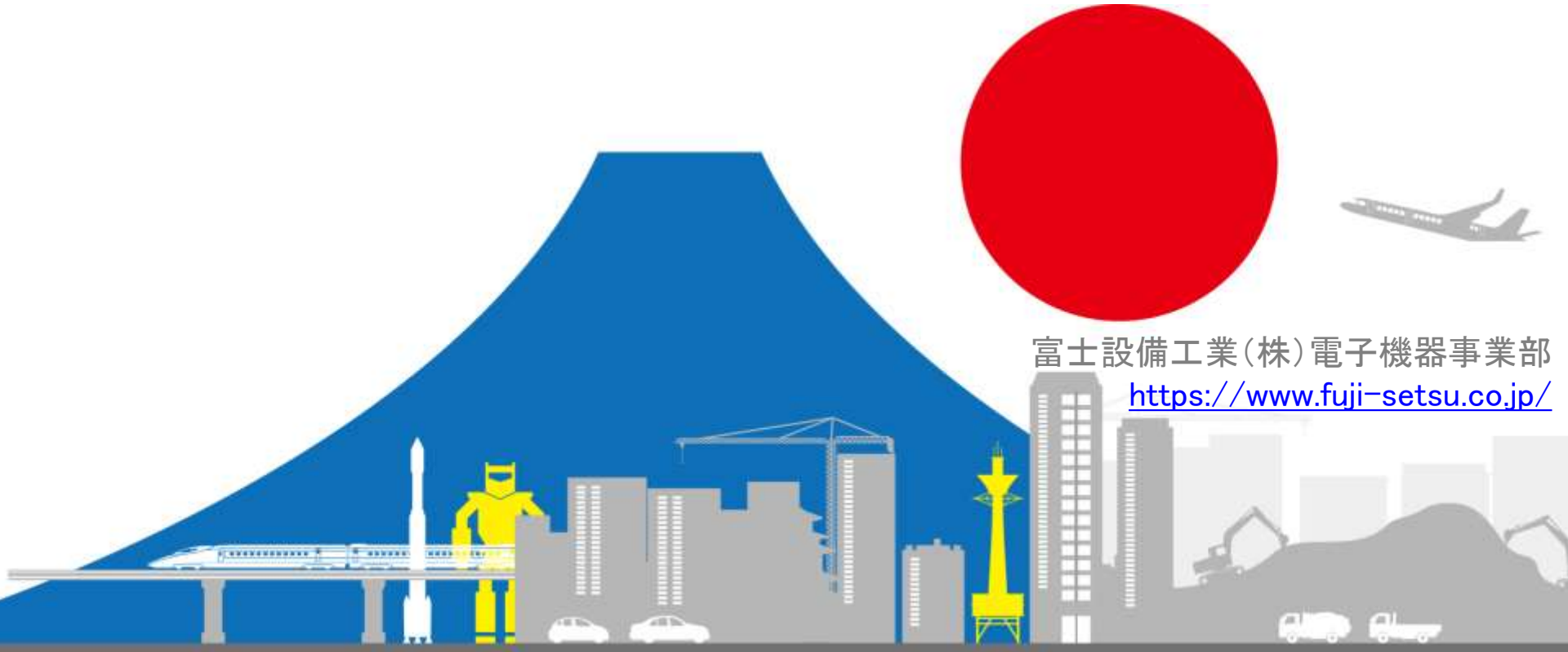
- ABI-Tester 機能でABIの規則に準拠
  - 他のサプライヤーの製品と繋がることで問題が発生しても、お相手の問題として切り分けすることができる。
  各社がSuperTestを使ってくれると、もっと助かるのだが、、

コンパイラユーザ成功例

- ライブラリのテスト
  - C99 なのにC89のライブラリが混同されていて、sin, cosなどの数学関数で問題になるところを、事前に回避できた。
  - 自動運転などで画像認識等はC++のライブラリに依存することになるので、SuperTestのライブラリテスト機能にも注目している。またロードマップ内のFloat16のテストや、AUTOSAR C++14 対応にも期待している。

- Depth Suite 機能
  - アーキテクチャ固有のデータ長（例えば char も int も24ビット）に合わせて演算精度のテストができる。

FUJI SETSUBI



富士設備工業(株)電子機器事業部

https://www.fuji-setsu.co.jp/