

DevSecOps for FuSA Applications – Best Practices for Development of Critical Software

DevSecOps – SDV、Fusa アプリケーション開発のベストプラクティス

■ 開発者の視点

- 「終わりのある開発」からOTAで更新し続ける「終わりの無い開発」へ、、

(出典: [Honda Stories](#))

■ 技術面の課題

- ソフトウェア開発工数の爆発
- 安全なソフトウェア更新(OTA)技術

(出典: [Open SDV Initiative](#))

■ 開発スピード向上がカギ

■ DevSecOps

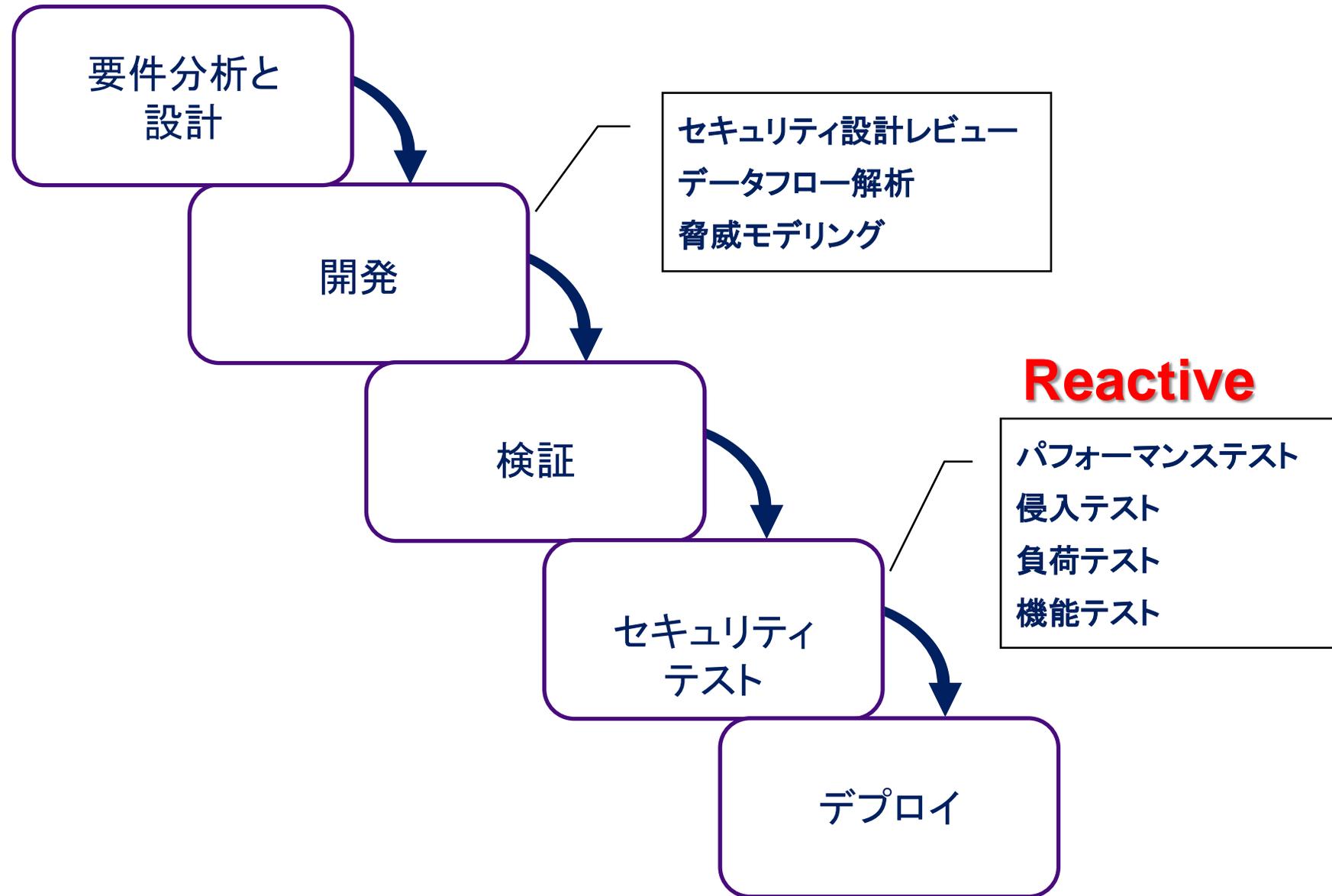
- 航空・宇宙業界の大規模開発にも採用
- 機能安全、セキュアコーディング、検証ツール
- デジタルツイン、デジタルスレッド

The LDRA logo is displayed in a white, bold, sans-serif font in the top right corner of the slide. The background of the slide features a network of thin, light-colored lines and circles, with a color gradient transitioning from light yellow on the left to dark orange on the right.

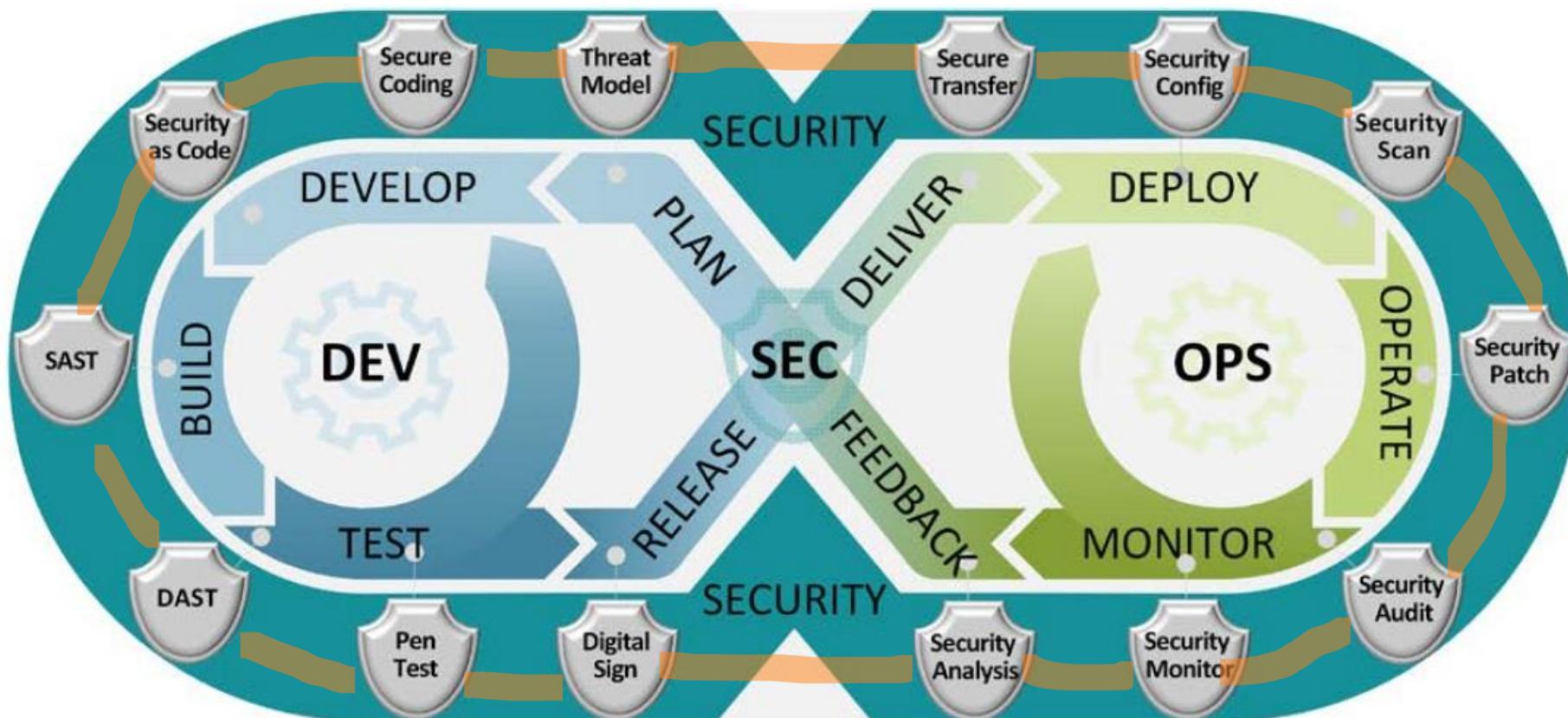
LDRA

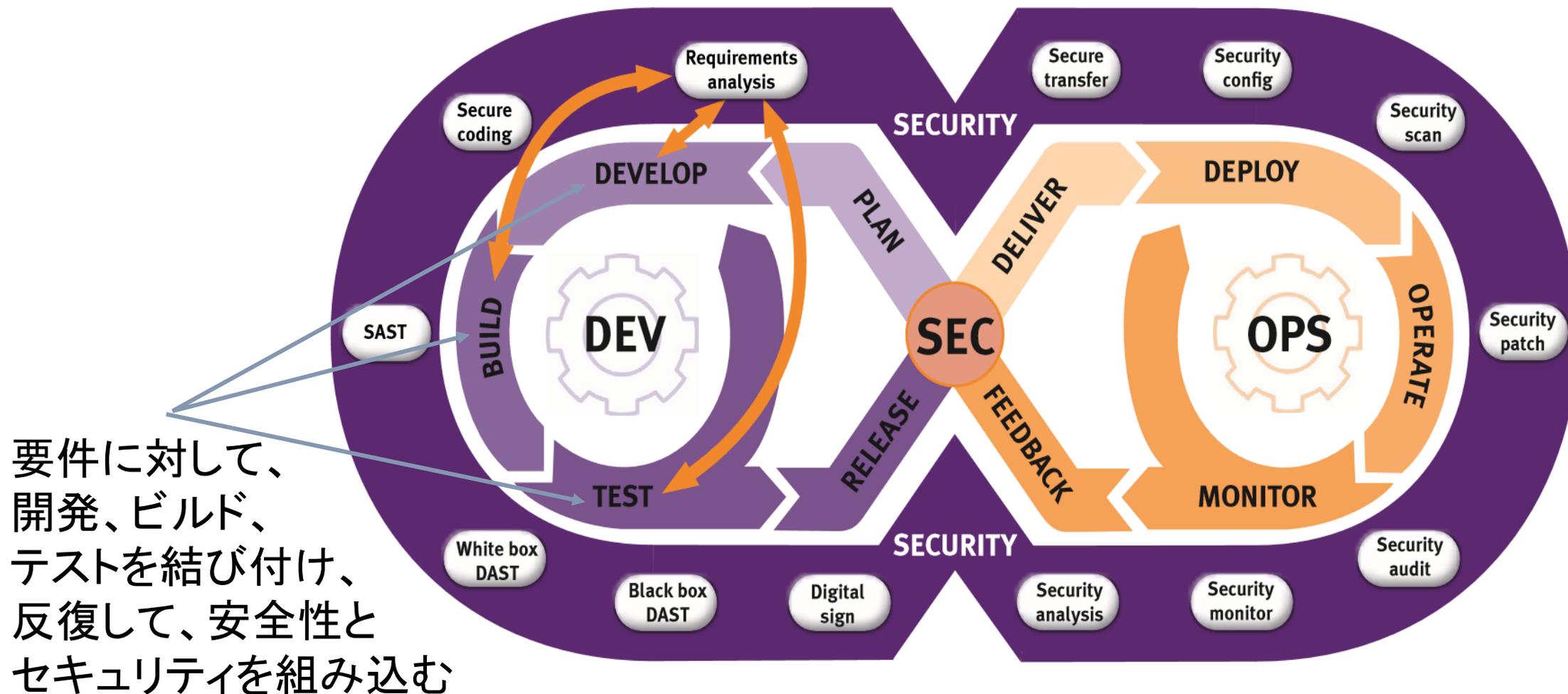
DevSecOps

シフトレフト、セキュアソフトウェア開発ライフサイクル



- ソフトウェア開発 (Dev)、セキュリティ (Sec)、情報技術運用 (Ops) を組み合わせて成果を安全にし、開発ライフサイクルを短縮する一連のソフトウェア開発プラクティス 出典: [米国国防総省のDevSecOpsイニシアチブ](#)



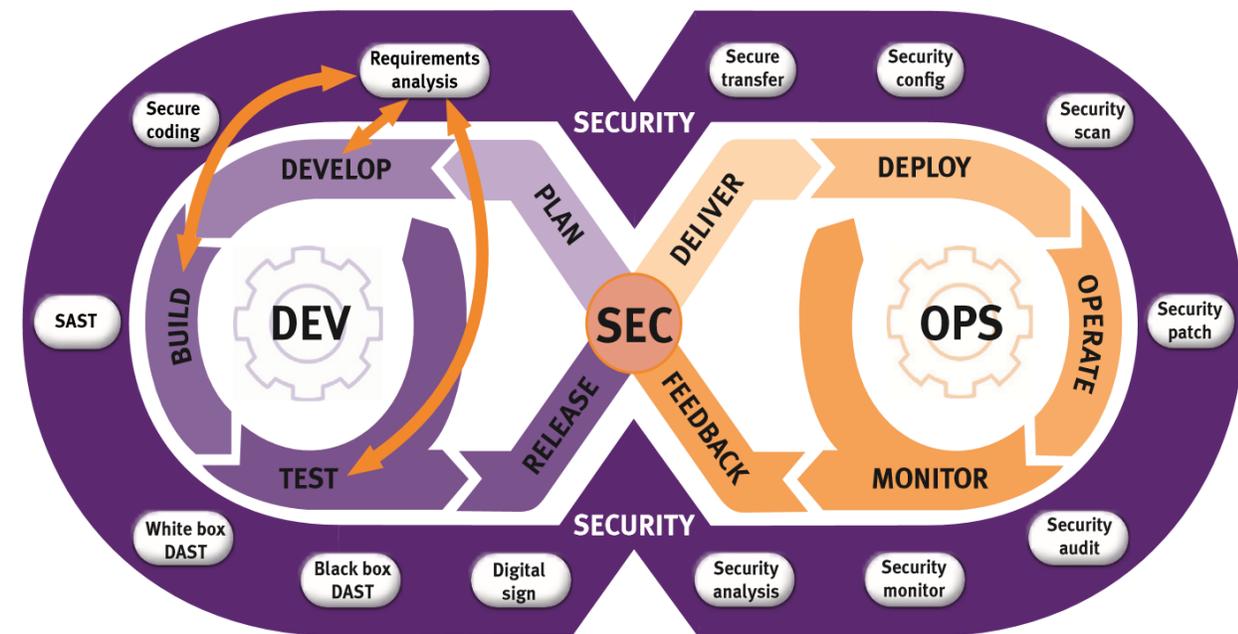


Shift left. Design security into your code.

<https://ldra.com/ldra-blog/shift-left-design-security-into-your-code/>

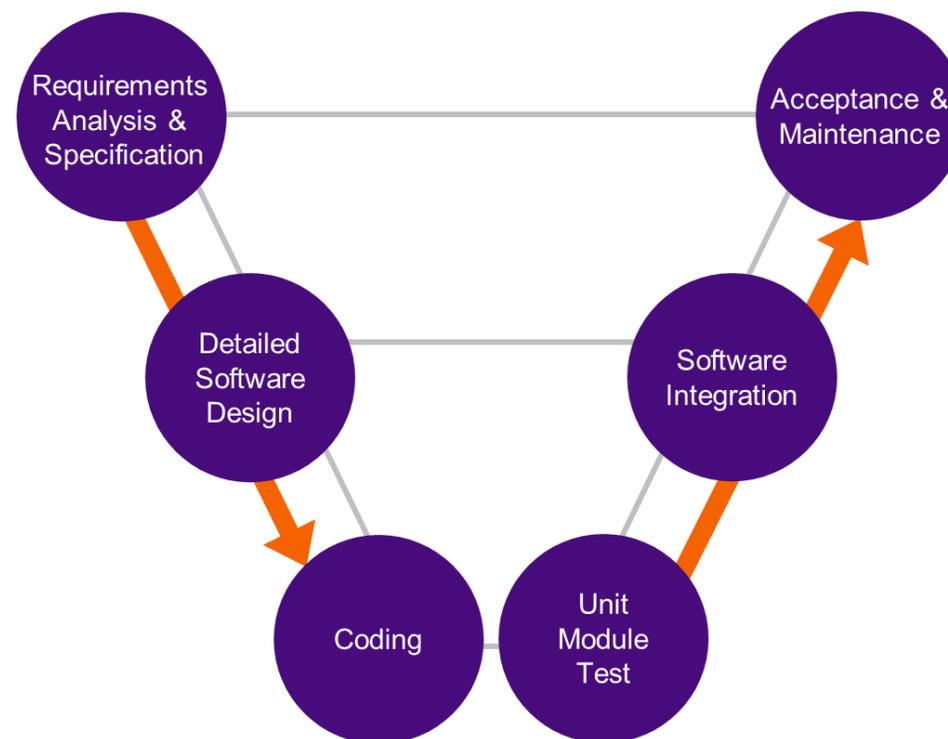
脆弱性を設計上排除する、もし検出された場合にはタイムリーかつ徹底的に対処される実用的な手段を備える

- 最初に機能要件とセキュリティ要件を確立
- セキュアなコーディング規約
- 早期にかつ頻繁にテスト
- 双方向の要件トレーサビリティ
- セキュリティ規格
- SAST(静的)およびDAST(動的)
セキュリティテストプロセスの自動化
- テイント解析で防御メカニズムをテスト



安全性の考慮をシステム、製品、プロセス設計の初期段階に統合し、安全が設計理念の本質的部分となるようにする

- 最初に機能要件と安全要件を確立
- 安全なコーディング規約
- 早期にかつ頻繁にテスト
- 双方向の要件をトレーサビリティ
- 安全を重視したプロセス規格
- テストプロセスの自動化
- 認定された検証ツールを使用





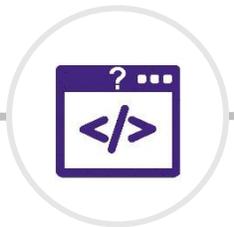
要件トレーサビリティ



規格が定めるオブ
ジェクティブの達成



コーディング規約
への準拠



コードの複雑性
の解析



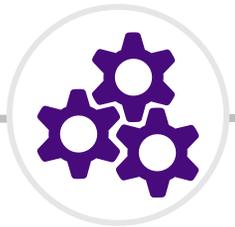
データフロー、
コントロールフロー解析



構造化カバ
レッジ解析



ターゲットレベル
でのテスト実行



ツール認定



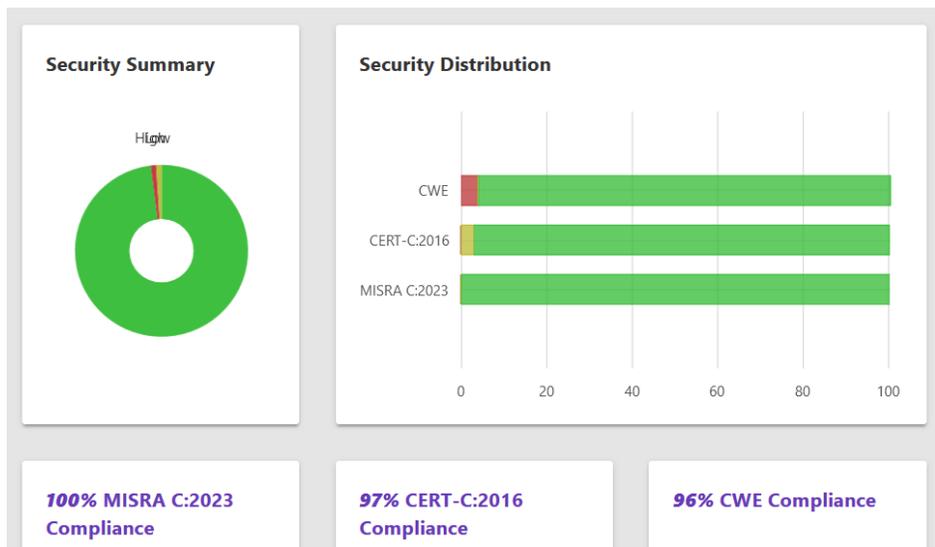
規格準拠の管理



LDRA tool suite

■ 最新のコーディング規約、言語機能

- MISRA C++:2023 & CWE-4.14
- Threading & multicore support
- Guideline recategorization
- C++17, including Lambda operation restrictions



LDRA tool suite MISRA C++:2023 Compliance Overview Report
System Set: Cpp_tunnel_lighting_system

69% MISRA C++:2023 Compliance 109 / 157 Guidelines	48 Guideline Violations	1 Mandatory 29 Required 18 Advisory 0 Document	852 Total Violations	11 Mandatory 378 Required 463 Advisory 0 Document
---	-----------------------------------	---	--------------------------------	--

Date of Analysis: Mon Sep 2 2024 12:24:13 | Report Produced on: Tue Sep 3 2024 09:21:28 | LDRA Version: 10.3.0 | Reporting Scope: Source File and Associated Header

MISRA C++:2023 Guidelines

Guideline	Compliance	Level	Violations	Description
1024	Not Compliant	Rule	1	Comparison of Incompatible Types
1025	Compliant	Rule	0	Comparison Using Wrong Factors
1037	Compliant	Rule	0	Processor Optimization Removal or Modification of Security-critical Code
1052	Not Compliant	Rule	36	Excessive Use of Hard-Coded Literals in Initialization
1056	Compliant	Rule	0	Invokable Control Element with Variadic Parameters
1064	Compliant	Rule	0	Invokable Control Element with Signature Containing an Excessive Number of Parameters

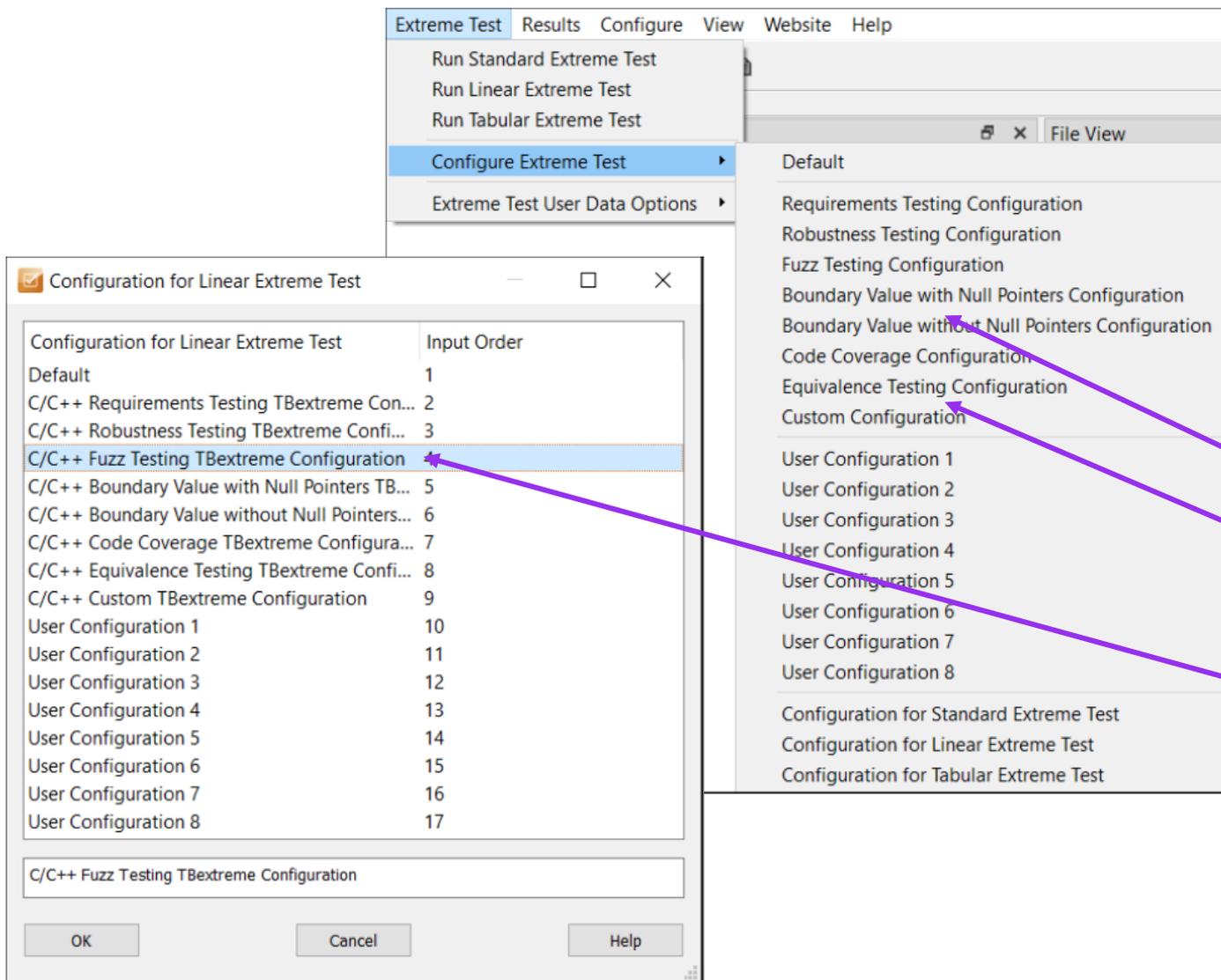
LDRA tool suite CWE-4.14 Compliance Overview Report
System Set: Safe_Uilities

90% CWE-4.14 Compliance 147 / 164 Guidelines	17 Guideline Violations	17 Rule 0 Checking 0 Optional 0 Informational	263 Total Violations	263 Rule 0 Checking 0 Optional 0 Informational
---	-----------------------------------	--	--------------------------------	---

Date of Analysis: Tue Sep 3 2024 09:19:18 | Report Produced on: Tue Sep 3 2024 09:19:40 | LDRA Version: 10.3.0 | Reporting Scope: Source File and Associated Header

CWE-4.14 Guidelines

Guideline	Compliance	Level	Violations	Description
1024	Not Compliant	Rule	1	Comparison of Incompatible Types
1025	Compliant	Rule	0	Comparison Using Wrong Factors
1037	Compliant	Rule	0	Processor Optimization Removal or Modification of Security-critical Code
1052	Not Compliant	Rule	36	Excessive Use of Hard-Coded Literals in Initialization
1056	Compliant	Rule	0	Invokable Control Element with Variadic Parameters
1064	Compliant	Rule	0	Invokable Control Element with Signature Containing an Excessive Number of Parameters



- 単体テストケース自動生成
- 機能安全規格の堅牢性試験
 - 境界値
 - 同値分割
- セキュアコーディングのファジングテスト

The LDRA logo is positioned in the top right corner of the slide. It consists of the letters 'LDRA' in a bold, white, sans-serif font. The background of the slide features a network of thin, light-colored lines and circles, suggesting a digital or interconnected theme.

DevSecOps

顧客事例：航空・宇宙・防衛業界

■ 顧客事例 1:

- 課題: 静的解析と動的解析のワークフローは自動化されていなかった
- 解決策: コンテナ化とパイプラインを使用して、検証を自動化プロセスに組み込むことで、解析を頻繁に実行できるようにした

■ 顧客事例 2:

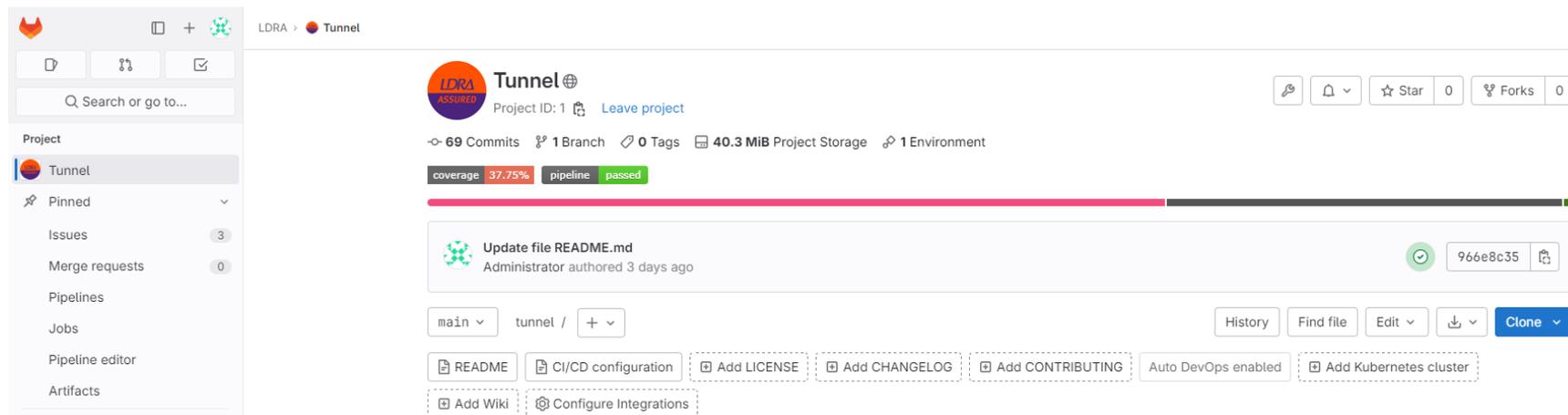
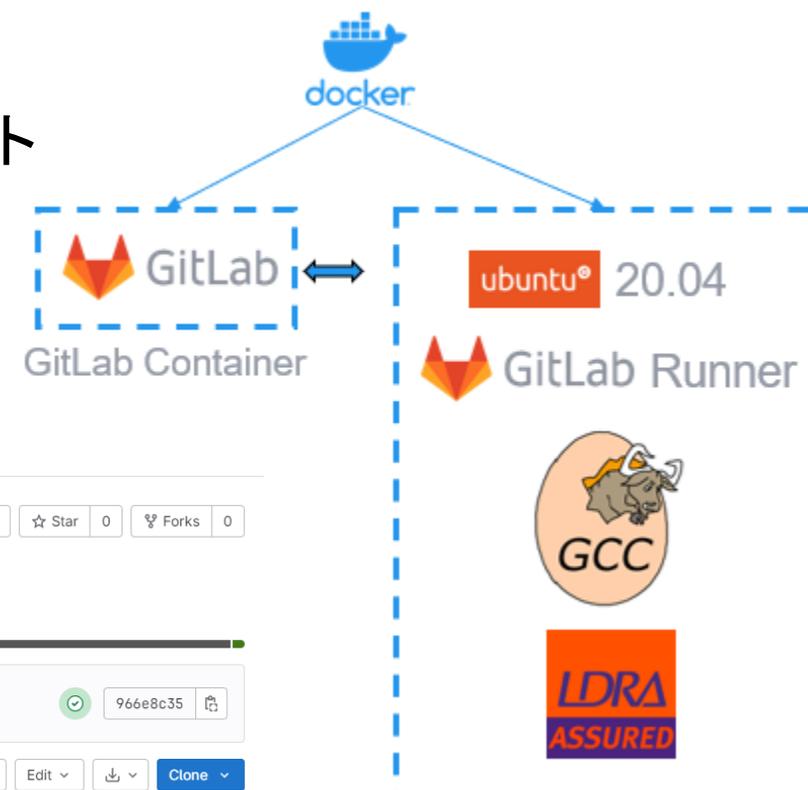
- 課題: 大規模なコードベースで静的解析、動的解析に長時間費やしていた
- 解決策: 複数のコンテナを並列実行し解析時間を最大6倍速めることに成功



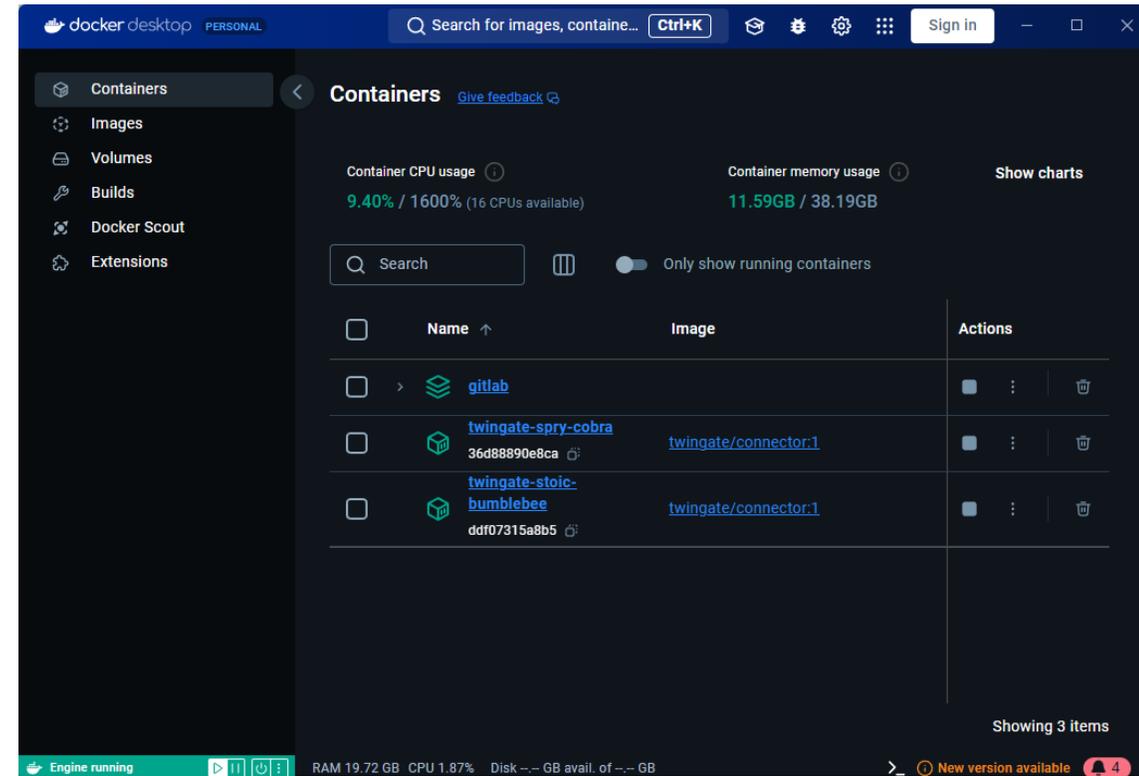
Domain	CPPs	CI/CD (no container)	Server tbwrkfls(8x licenses)	Localized tbwrkfls(8x licenses)	Status
App1	64	3h:37m	36m	24m	All pass
App2	30	1h:19m	13m	9m	All pass
CPI1	126	6h:59m	1h:19m	49m	All pass
CPI2	44	1h:28m	16m	11m	All pass
CPIX	61	2h:28m	37m	20m	All pass

■ GitLab を採用

- 静的テストと単体テストを並列実行
- 10,000 以上の TCF (テストケースファイル) で回帰テスト
- チェックインから結果までのシームレスなワークフロー



- Gitlab と LDRA は別のコンテナで実行
- 共有ストレージは、コードカバレッジ用にインストールされたソースを共有し、レポートを取得するために使用
- 大規模な並列化が可能になり、8 コアマシンで解析時間を最大6分の1に短縮することに成功
- LRU(列線交換ユニット)だけを検討する領域から、航空機全体のみより大規模なシステムを検討する領域に



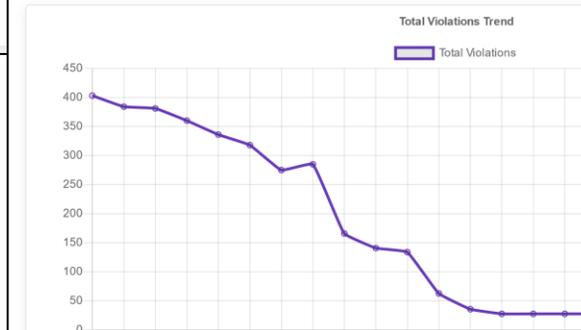
LRU (Line replaceable unit) : 自己診断装置で異常が見られた場合に即座に取替え可能な装置

顧客事例 2: 大規模検証成果物を管理

The screenshot shows the LDRA Projects overview page. It features a navigation bar with 'HOME', 'LDRAVAULT', 'LCMS', and 'ADMIN'. Below the navigation, there are three project cards: 'Cpp17_Cashregister', 'STM32G031K8_SelfBalance', and 'RM48L8950_Dice'. Each card includes a brief description and a 'View Components' button. The 'STM32G031K8_SelfBalance' card is highlighted, showing it has 1 component.

Manage multiple projects...

The screenshot shows the LDRA Components page for the 'STM32G031K8_SelfBalance' project. It features a navigation bar with 'HOME'. Below the navigation, there are three tabs: 'Overview', 'Summary', and 'Project Details'. The 'Overview' tab is selected, showing a list of components. One component, 'STM32G031K8_SelfBalance_HAL', is highlighted, showing it has 1 upload.



...view trends...

The screenshot shows the LDRA TBmanager Project Report for the 'STM32G031K8_SelfBalance' project. It features a navigation bar with 'HOME'. Below the navigation, there is a 'Project Tree' on the left and a 'Project Report' on the right. The report shows a table of violations with columns for SLR, HLR, HLT, LLR, and LLT. The table contains several rows of violations, including 'R [SYS_100] The system shall have a set of safe utilities' and 'R [HLR_100] Safe routines to convert integers to string'.

multiple components...

...and generate reports

■ DO-178C や ISO 26262 など規格への準拠を集約するLCMS

Document Number	Document Title	Rev	Rev Date	SCM	SOI	Checklist	Pass	Comments	Open	Closed
800-PSAC-01	Plan for Software Aspects of Certification	C	31-Oct-2024 04:55 PM	CC1	1	View / Modify	Pass	View	2	0
800-SAS-01	Software Accomplishment Summary	A	31-Oct-2024 04:55 PM	CC1	1	View / Modify	Pass	View	0	0
800-SCS-01	Software Code Standards	1.3	31-Oct-2024 04:55 PM	CC2	1	View / Modify	Pass	View	1	0
800-SCI-01	Software Configuration Index	C	31-Oct-2024 04:55 PM	CC1	1	View / Modify	Pass	View	0	0
800-SCMP-01	Software Configuration Management Plan	A	31-Oct-2024 04:55 PM	CC1	1	View / Modify	9 / 14	View	1	0
800-SDD-01	Software Design Document	A	31-Oct-2024 04:55 PM	CC2	1	View / Modify	Pass	View	0	0
800-SDS-01	Software Design Standards	A	31-Oct-2024 04:55 PM	CC2	1	View / Modify	4 / 6	View	0	0
800-SDP-01	Software Development Plan	D	31-Oct-2024 04:55 PM	CC1	1	View / Modify	Pass	View	0	0
800-SECI-01	Software Life Cycle Environment Configuration Index	A	31-Oct-2024 04:55 PM	CC2	1	View / Modify	Pass	View	0	0

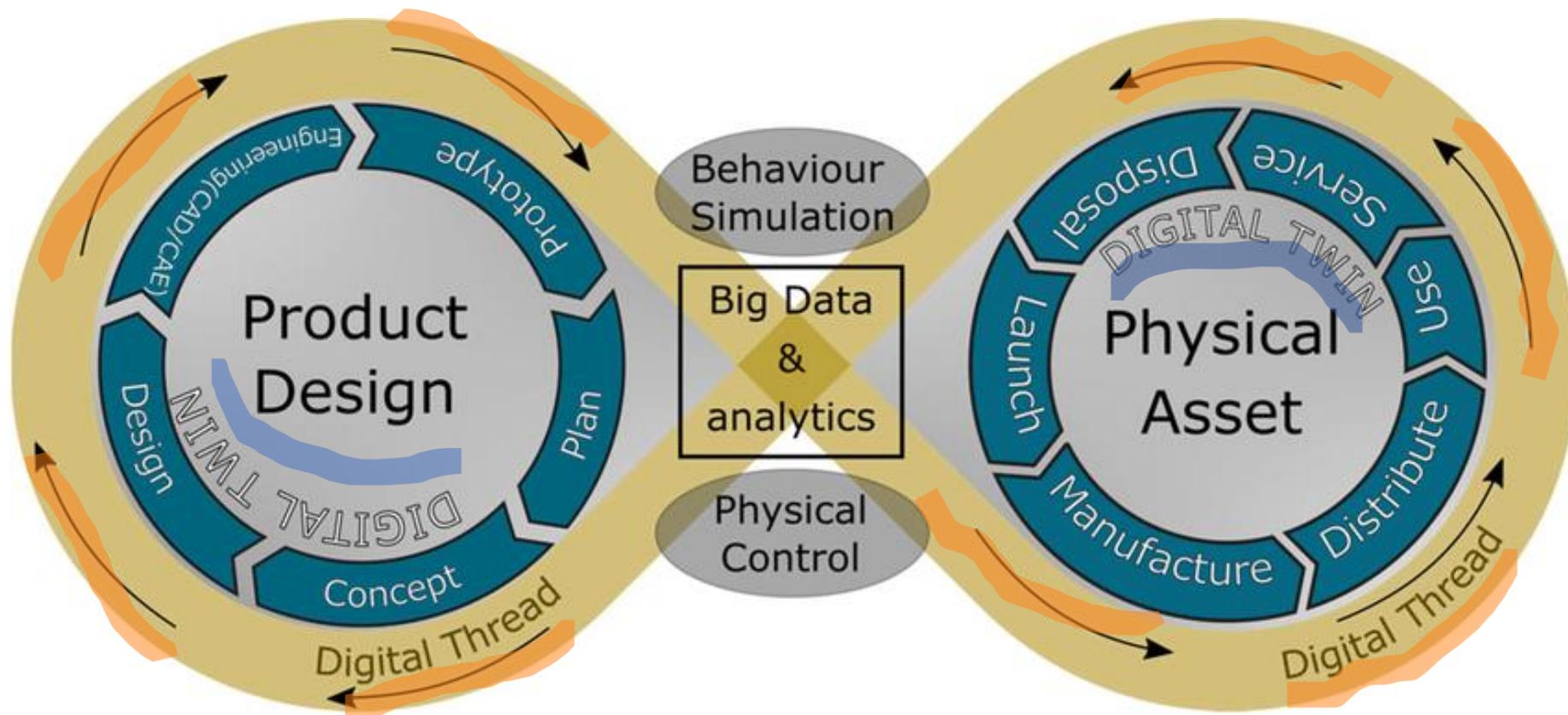
The LDRA logo is positioned in the top right corner of the slide. It consists of the letters 'LDRA' in a bold, white, sans-serif font. The background of the slide features a network of thin, light-colored lines and nodes, with a bright, glowing light source on the left side that creates a lens flare effect across the top half of the image.

LDRA

DevSecOps

デジタルスレッド、デジタルツイン

- デジタルスレッド：一貫性のあるデータを共有して有機的につなぐ
- デジタルツイン：現実世界のモノを仮想空間上に再現
- 両者により継続的な設計変更や最適化の自由度を高める



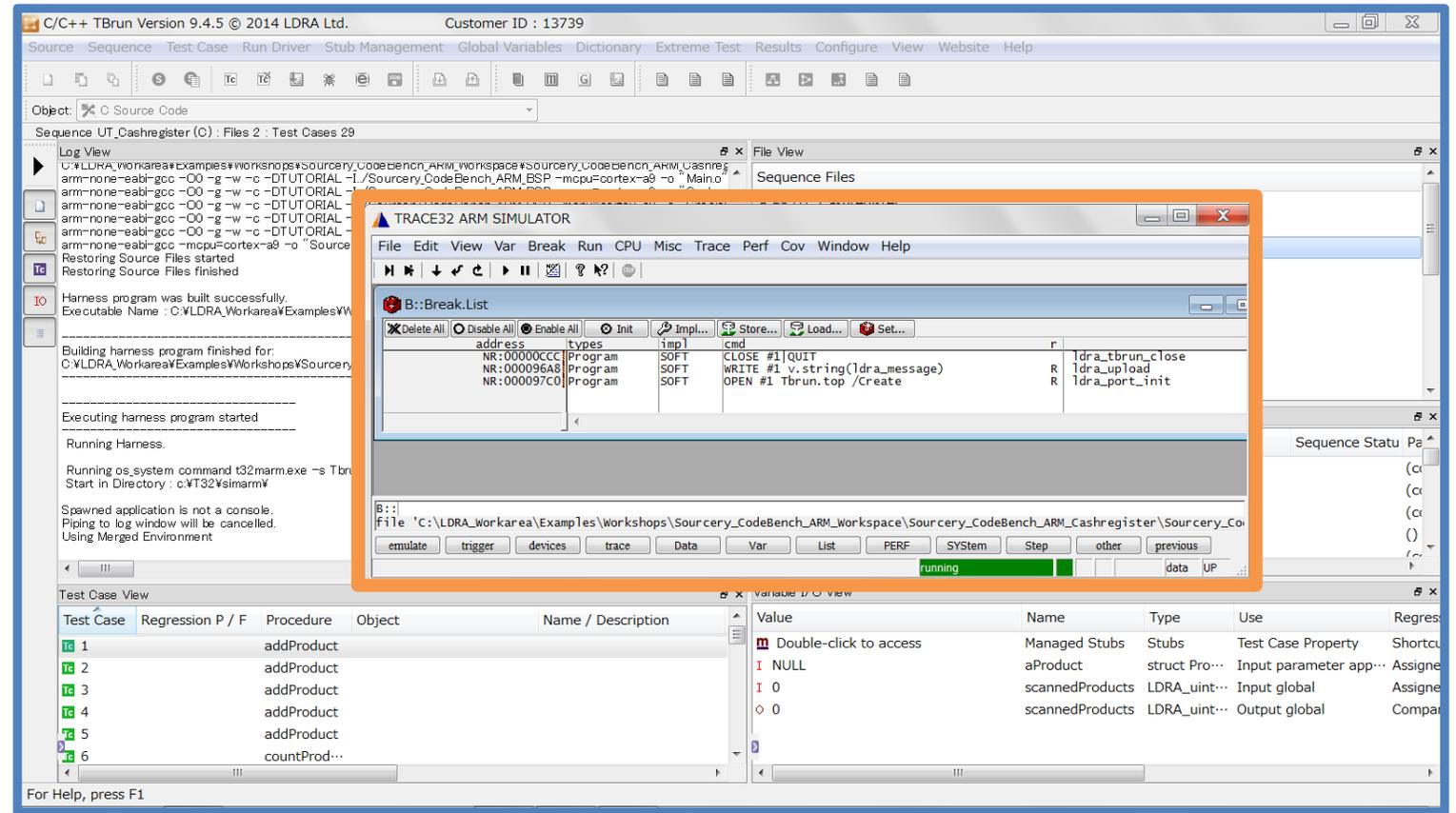
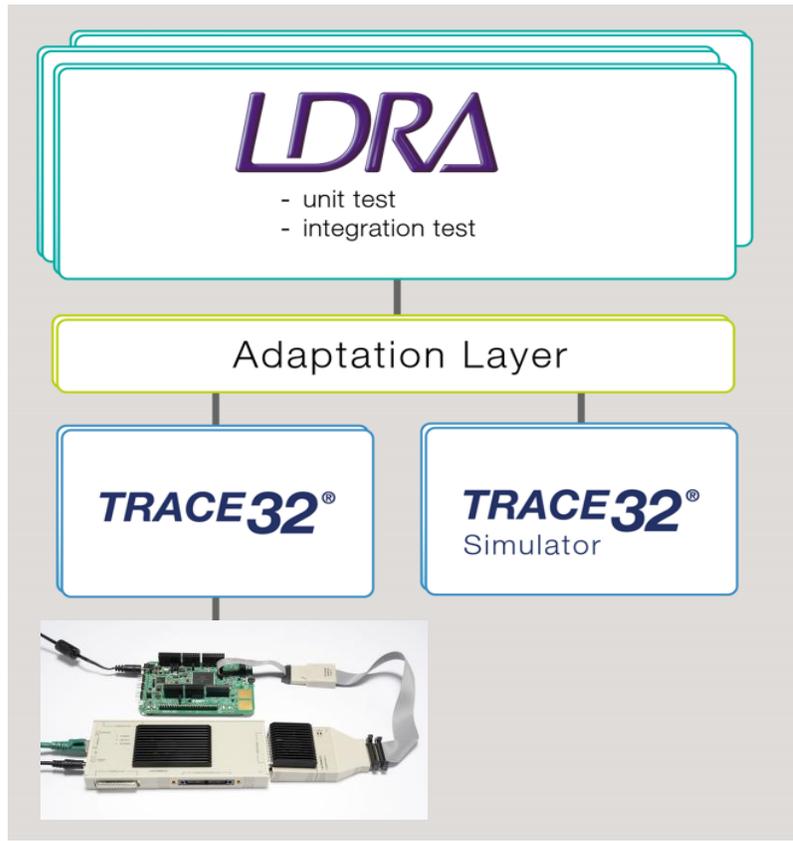
- デジタルスレッド: MBSEでCAD、モデル、コードをつなぐ
- デジタルツイン: モデルシミュレータ、インストラクションセットシミュレータ

Integrating systems modeling with CAD

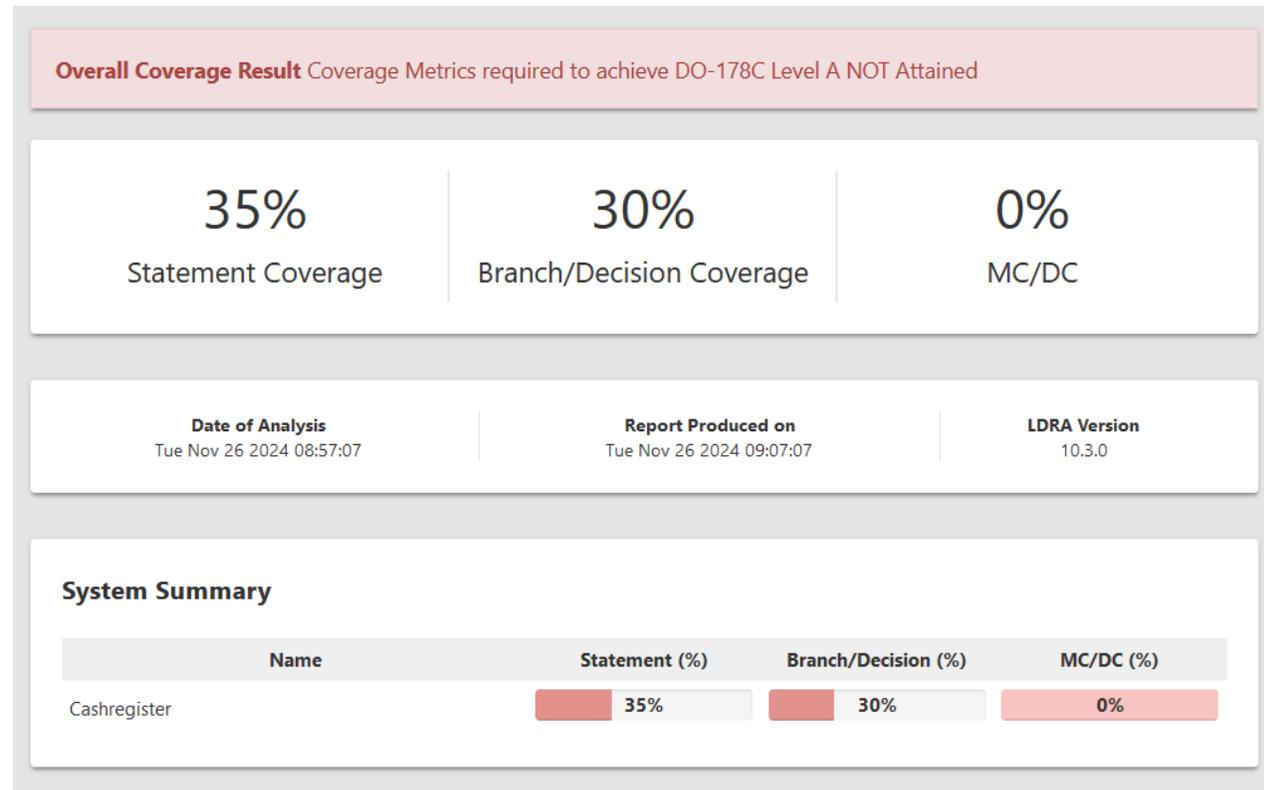
The screenshot displays a software interface for integrating systems modeling with CAD. It features three main panels: 'SysML Model' on the left, 'Connection Type' in the center, and 'Windchill 9 Repository' on the right. The 'SysML Model' panel shows a hierarchical tree structure with 'UnmannedAerialVehicle' expanded to show various components like 'auto1', 'comm_con1', 'databus', 'flight_con1', 'gprs1', 'gps', 'pltfrm1', 'pyld_cont', 'ssr1', 'sw', 'therm1', 'therm2', 'vis1', 'wifi1', 'Video_Camera_I_F', and 'VisualCamera'. The 'Connection Type' panel has 'Reference' selected, with other options like 'Function Wrap', 'Data Map', 'Model Transform', and 'Composite'. The 'Windchill 9 Repository' panel shows a 3D CAD model of a UAV fuselage with a red arrow pointing from the 'pyld_cont' element in the SysML Model to the 'Reference' connection type. Below the CAD model, there is a list of components including 'vis1: VisualCamera (A.1)' and 'wifi1: WiFi_WiMax_Module (A.1)'. A text box at the bottom of the screenshot reads: 'Use Case 1 – If a SysML element is connected to a CAD model, SEs can visualize the CAD model in SysML'.

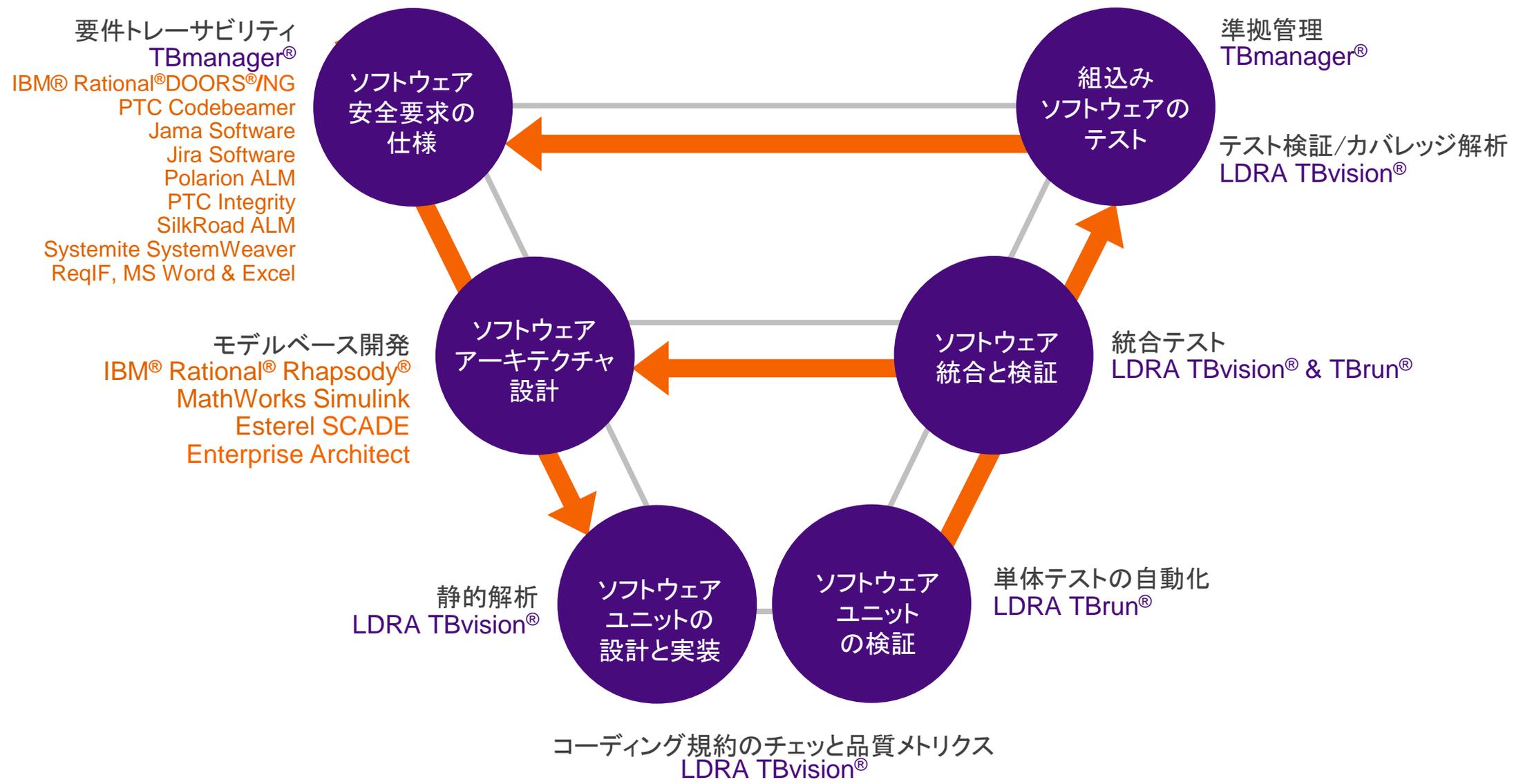
Use Case 1 – If a SysML element is connected to a CAD model, SEs can visualize the CAD model in SysML

- あらゆるコンパイラ、デバッガ、実行環境での検証作業をサポート



デジタルツインにおいても、要件ベーステストの十分性は構造化カバレッジやデータフローカバレッジ解析で評価され、デジタルスレッドの一部となる





TBmanager 要件～テストを双方向にトレース

System Requirements Software High-Level Requirements Software Low-Level Requirements Source Code

Relationships

(0) Three-Level Requirements to Mappings

Select None (13) Requirements 1 Select None (34) Requirements 2 Select None (58) Requirements 3 Select None (47) Mappings

- SYS_0010, Display , (2 Notes)
- SYS_0020, Initialisation and configuration , ...
- SYS_0030, Output Calculation , (1 Note)
- SYS_0040, Photometer, (1 Note)
- SYS_0050, Cleanliness factor
- SYS_0060, Lighting control unit , (1 Note)
- SYS_0070, Luminaries
- SYS_0080, Sirens and Signs, (1 Note)
- SYS_0090, Failed Power Supply
- SYS_0100, Lighting Adjustment
- SYS_0110, Lamp output units
- SYS_0120, Lamp sizes , (1 Note)
- SYS_0130, Required luminance at ground level

The Tunnel Lighting system shall be configurable via an external file and take into account tunnel dimensions, zones, spacing for signs ,and efficiency factor

Requirement Body

- HLR_0010, Starting display software, (1 Note)
- HLR_0020, Input option photometer nominal ran...
- HLR_0030, Input options photometer input out ...
- HLR_0040, Input options EXIT
- HLR_0050, Input options days since cleaning no...

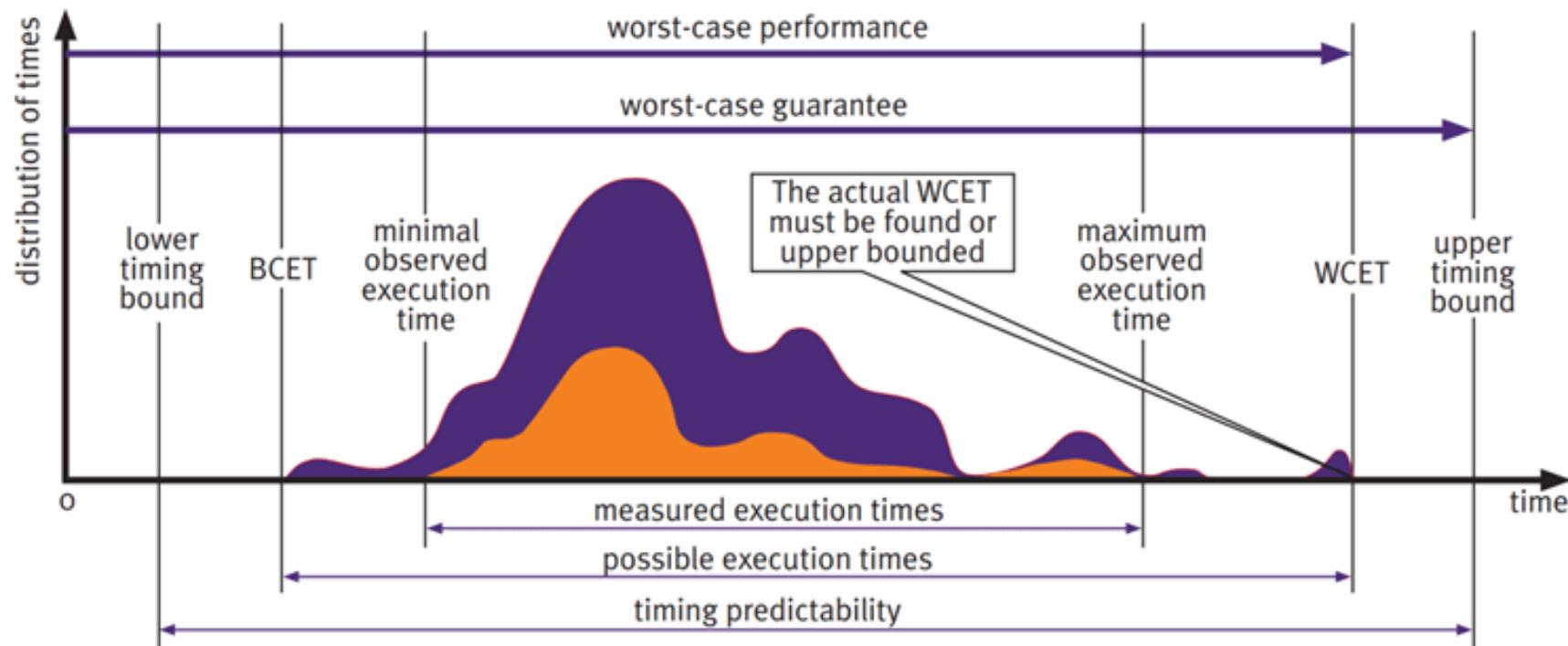
software, the try the user for appropriate data for escribed in the erview.

- LLR_0010, Instantiate Cell
- LLR_0020, Initialise Cell , (1 Note)
- LLR_0030, Set Emergency output level
- LLR_0040, Set PoweredOutputLevel , (1 Note)
- LLR_0050, Cal
- LLR_0060, Get
- LLR_0070, Get
- LLR_0080, Get
- LLR_0090, Get
- LLR_0100, Get
- LLR_0110, Get
- LLR_0120, Init
- LLR_0130, Set
- LLR_0140, Get

A Cell shall be inst maximum lumens, and zero for the ce

- Bool TunnelData::Cell::InitialiseCell(const Sint_32 Lu...
- Bool TunnelData::DataIn::GetData(TunnelData::Tun...
- Float_64 TunnelData::Cell::CalculateCellOutput(Floa...
- Float_64 TunnelData::Lamp::GetMaximumLumens();
- Float_64 TunnelData::Lamp::GetMinimumLumens();
- Float_64 TunnelData::LampType::GetMaximumLum...
- Float_64 TunnelData::LampType::GetMinimumLum...
- Float_64 TunnelData::SystemData::GetEmergencyL...
- Float_64 TunnelData::SystemData::GetLampMaxim...
- Float_64 TunnelData::SystemData::GetLampMinim...
- Float_64 TunnelData::SystemData::GetSoilingFacto...
- Sint_32 TunnelData::LampType::GetPowerRequired(...
- Sint_32 TunnelData::SystemData::GetDaysBetween...
- Sint_32 TunnelData::SystemData::GetExitSignSpaci...
- Sint_32 TunnelData::SystemData::GetLampPowerR...
- Sint_32 TunnelData::SystemData::GetSirenSpacing();
- Sint_32 main();
- TunnelData::Cell::Cell();

- デジタルツインでタイミング解析することでモデルが物理デバイスに対して忠実に機能することを確認できる



WCET: Worst-Case Execution Time
BCET: Best-Case Execution Time
ACET: Average-Case Execution Time

The LDRA logo is positioned in the top right corner of the slide. It consists of the letters 'LDRA' in a bold, white, sans-serif font. The background of the slide features a network of thin, light-colored lines and nodes, with a bright, glowing light source on the left side that creates a lens flare effect across the top half of the image.

LDRA

まとめ

システム開発・運用の一連の工程で発生する問題は後工程になるほど対策コストが増大

- DevSecOps
- シフトレフト
- 航空宇宙業界で実証済みの検証ツール
- 継続的インテグレーション
- コンテナ化
- デジタルツイン、デジタルスレッド



発射後4分 宇宙船を分離することができないまま上空で爆発し、失敗に終わりました。



(SpaceX)

しかし、ロケットが爆発したにもかかわらず、技術者たちは歓声を上げていたのです。



ギャレット・リースマン (スペースX社 上級顧問)

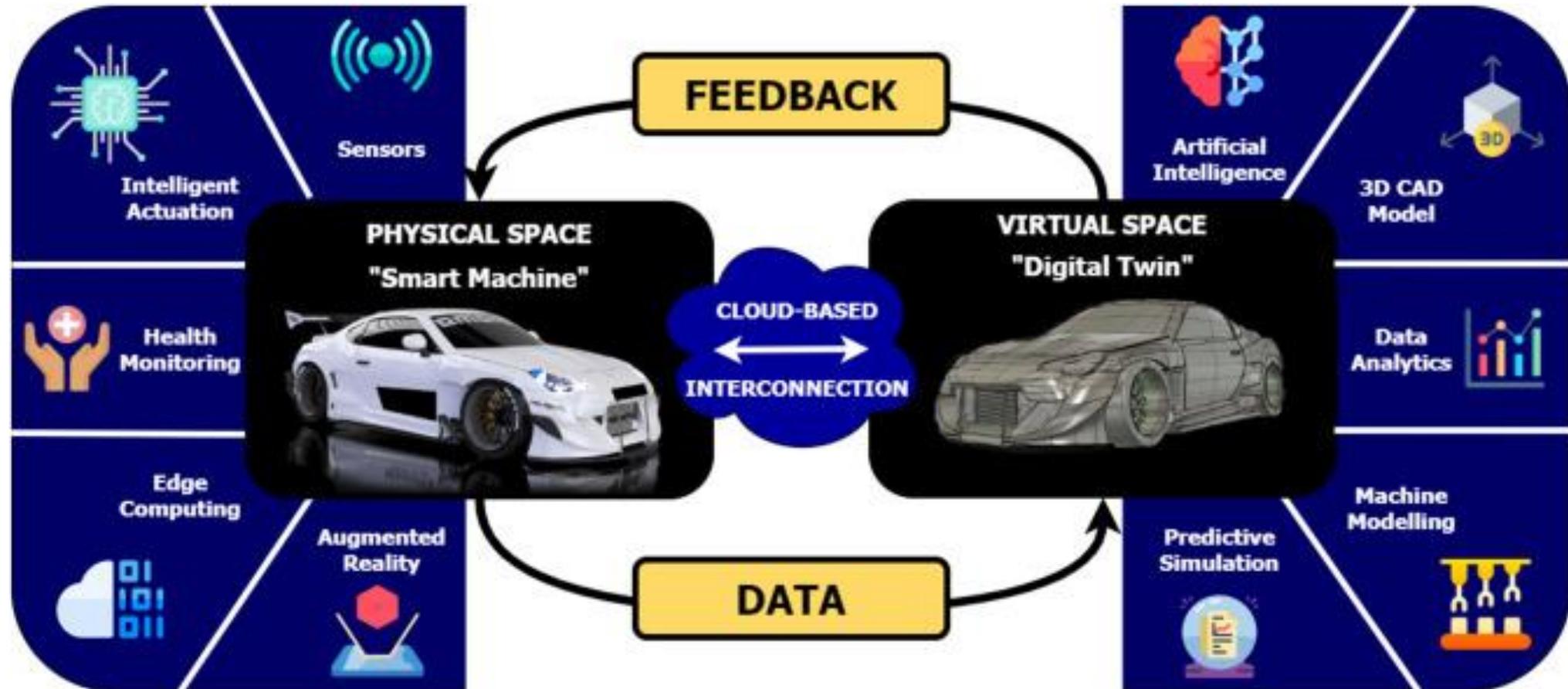
「ロケットが爆発したのにエンジニアたちが歓声を上げているというのは不思議に思うでしょう。なぜ歓声をあげるかというと、ロケットの爆発は本当の失敗ではなく期待以上の成果が得られたからなんです。確かにロケットはとても高価ですが、ロケットよりも高価なもの、それは時間です。スペースX社は大量のデータを集めるのを非常に得意としています。ロケットには多くの機器を載せており、そこから常にデータが送られてきます。独自に開発したAIツールなどを駆使して、すべてのデータを分析。迅速に問題を修正し、成功に導いています。つまり、大爆発したとしても、その過程で大事なことを学び、すぐに修正して次に進むことができるため、失敗ではなく成功と考えているのです。」

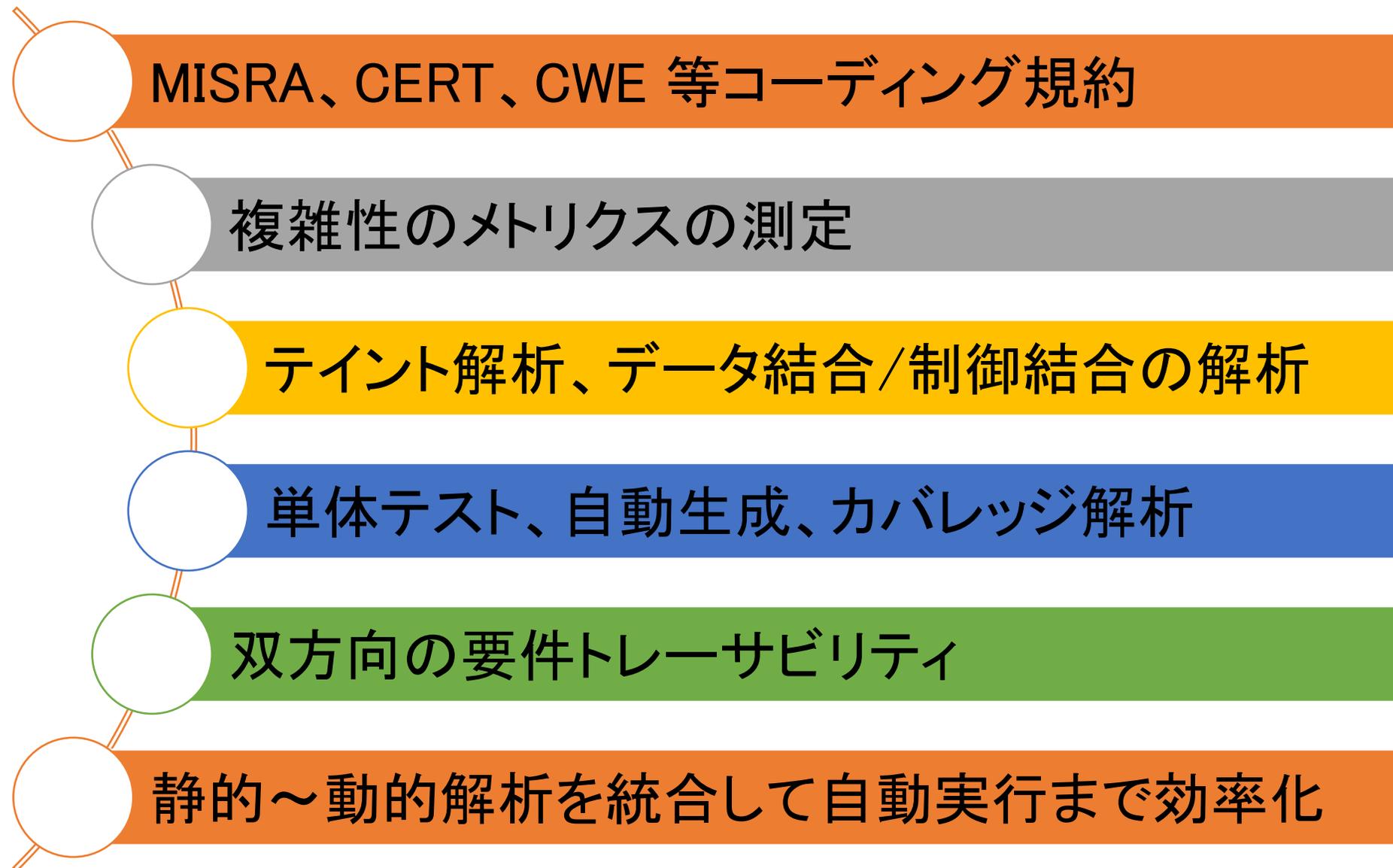
2024年2月26日 2回目の飛行試験 前回失敗した分離に成功

2024年3月14日 3回目の飛行試験 海上への帰還はならなかったが宇宙空間に到達

2024年6月6日 4回目の飛行試験 目標をすべてクリア 軟着陸に成功

失敗から1年足らずで成功までこぎつける。この驚異的なスピードこそ、コスト削減の上で大切だと言います。







LDRA Technology
Director – Field Engineering

Jay Thomas (ジェイ・トーマス)

セーフティクリティカルでミッションクリティカルなソフトウェアの構築と保守に重点を置いてキャリアを重ねて来た。これには、航空機や宇宙船、打ち上げロケットなど航空宇宙分野のソフトウェアのほか、産業制御や自動車へと分野を跨いだベストプラクティスの適用も含まれる。SpaceX 社の初期の従業員でもあり、現在、最も安全な軌道投入法である Falcon ロケットのフライトソフトウェアを開発した。現在、LDRA Technology 社のフィールドエンジニアリング担当ディレクターとして、組込み向けの検証の実践に取り組んでいる。



DevSecOps - Best Practices for V&V in the cloud

Jay Thomas, Director – Field Engineering, LDRA, USA

<https://ldra.com/events/devsecops-best-practice-for-vv-in-the-cloud-6pm-gmt/>



LDRA スタンダード認証支援テストツール
<https://www.fuji-setsu.co.jp/products/LDRA/>



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