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

Copyright © LDRA Limited 2025

SDV、DevSecOps、LDRA社検証ツール

- 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）

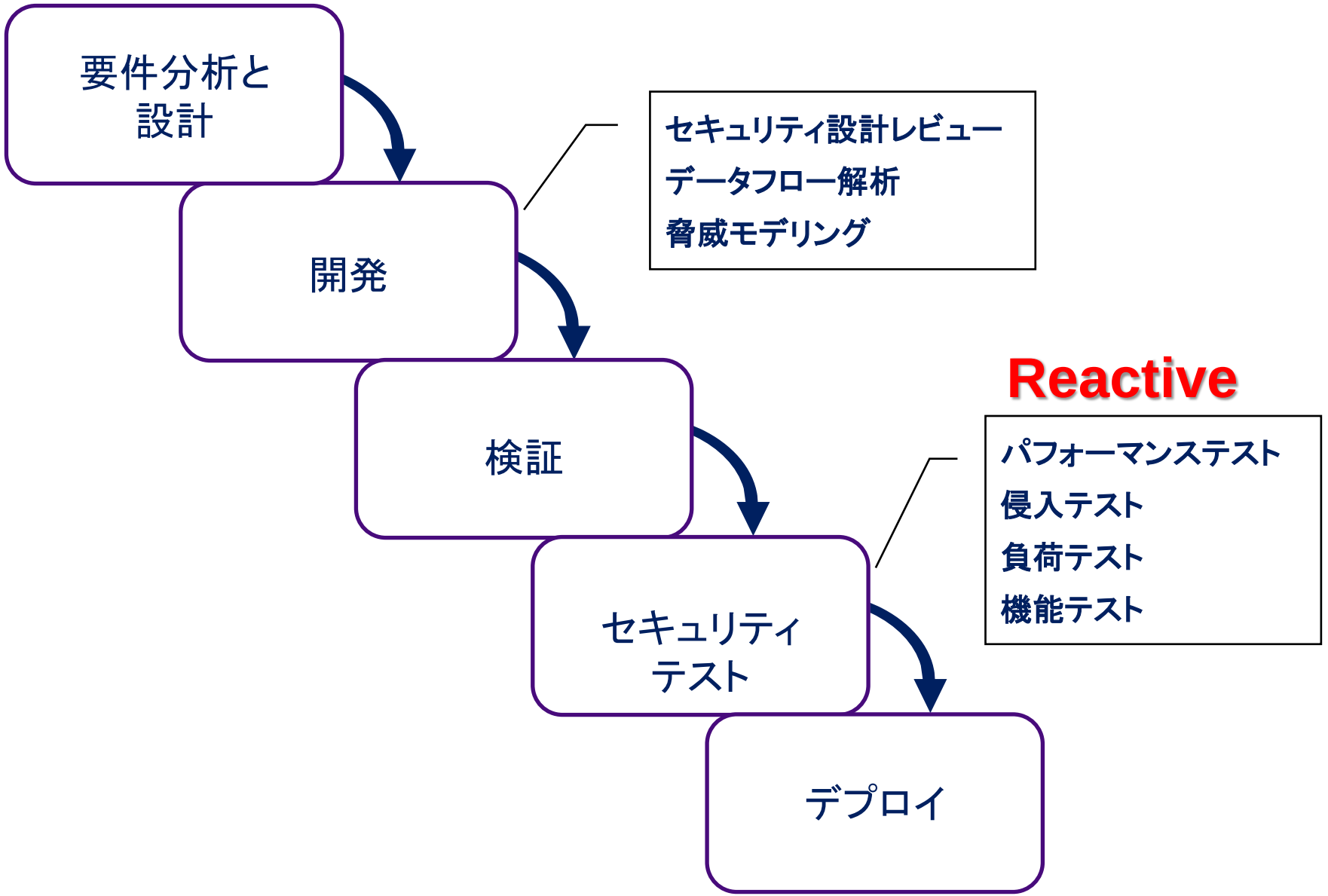
- 開発スピード向上がカギ
- DevSecOps
  - 航空・宇宙業界の大規模開発にも採用
  - 機能安全、セキュアコーディング、検証ツール
  - デジタルツイン、デジタルスレッド

LDRA

DevSecOps

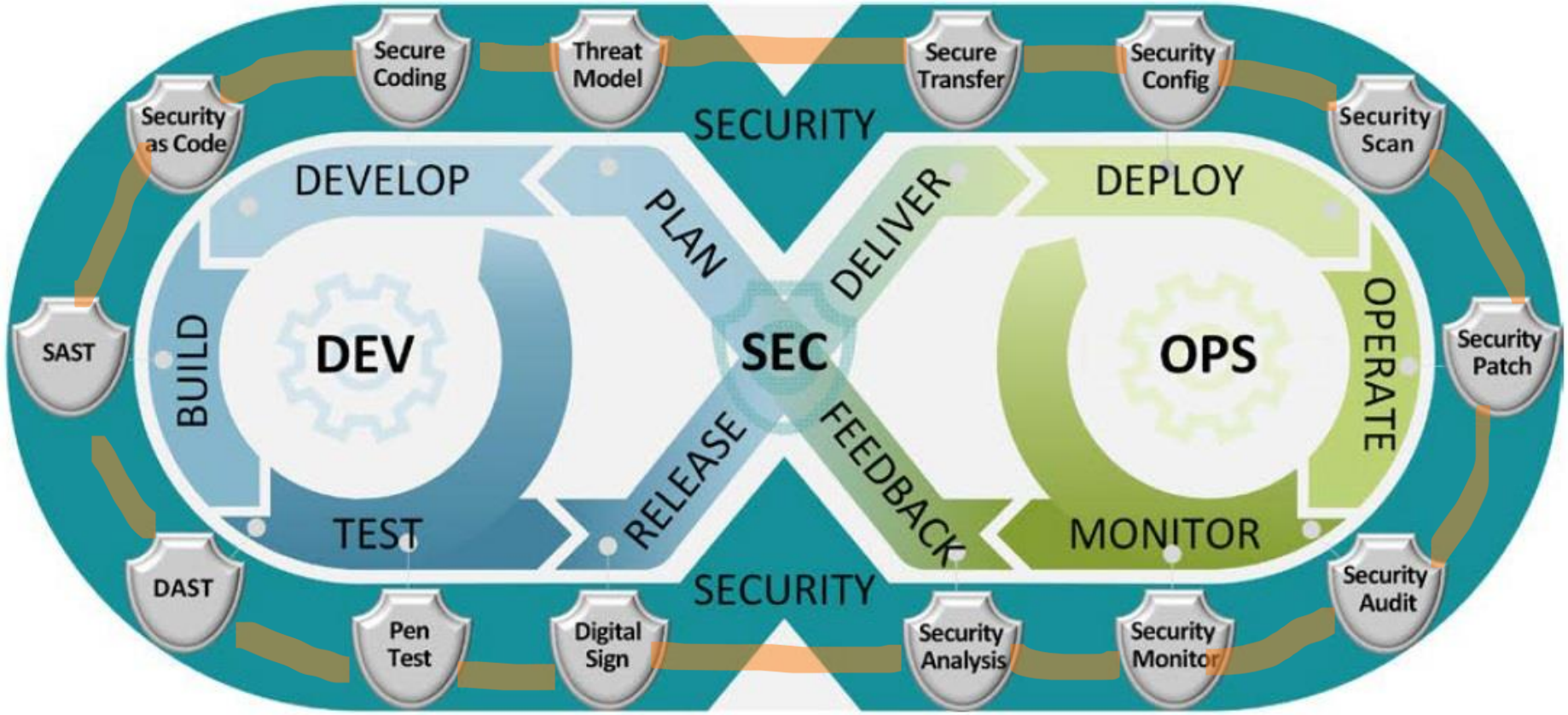
シフトレフト、セキュアソフトウェア開発ライフサイクル

サイバーセキュリティ：従来の取り組み

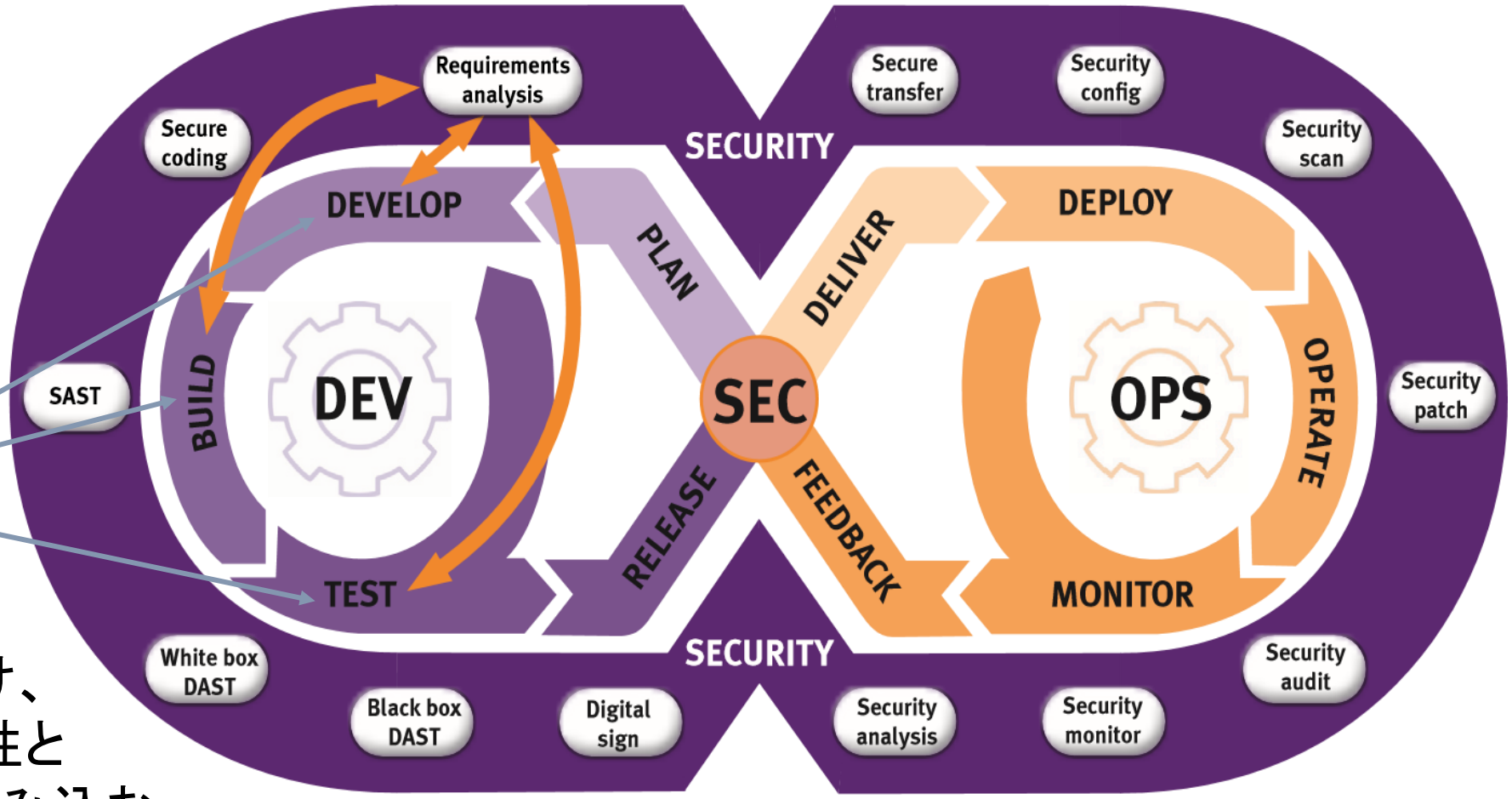


What is DevSecOps?

- 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



DevSecOps、機能安全系でのシフトレフト



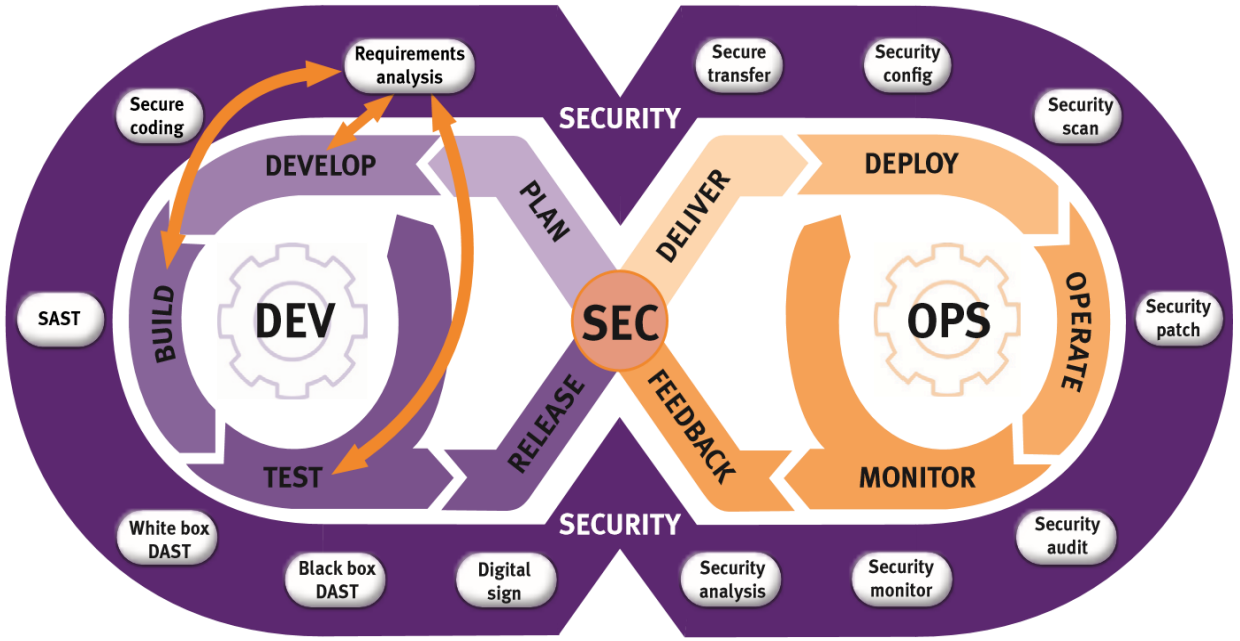
要件に対して、
開発、ビルド、
テストを結び付け、
反復して、安全性と
セキュリティを組み込む

Shift left. Design security into your code.
https://ldra.com/ldra-blog/shift-left-design-security-into-your-code/

シフトレフト、セキュリティバイデザイン

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

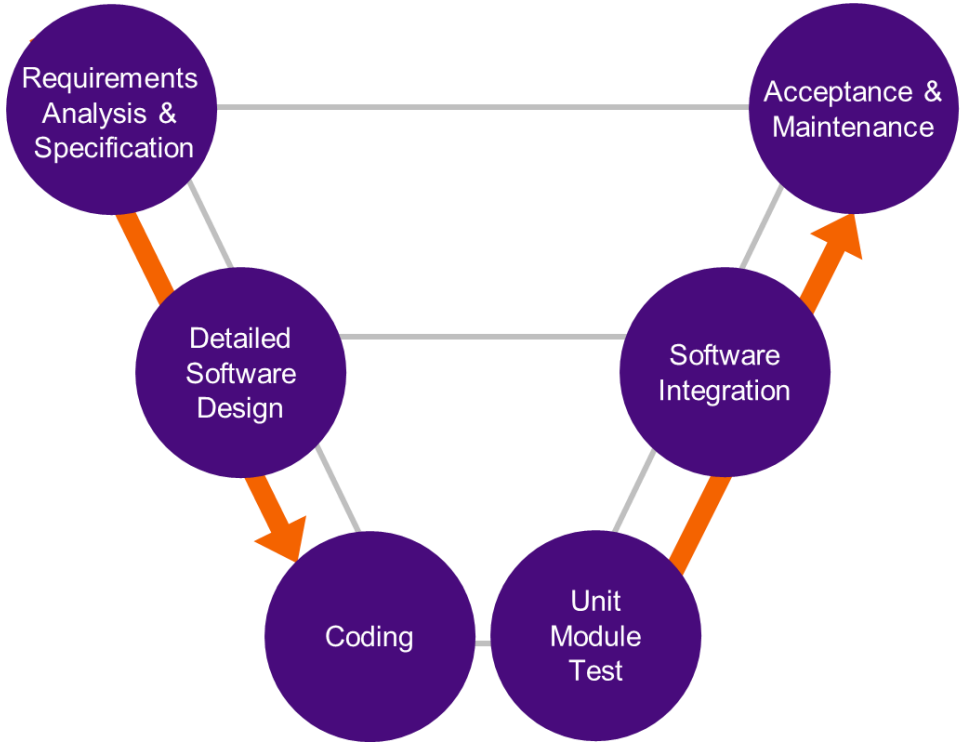


https://www.fuji-setsu.co.jp/products/LDRA/CyberSecurity.html

シフトレフト、セーフティバイデザイン

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



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

機能安全、セキュアコーディング

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

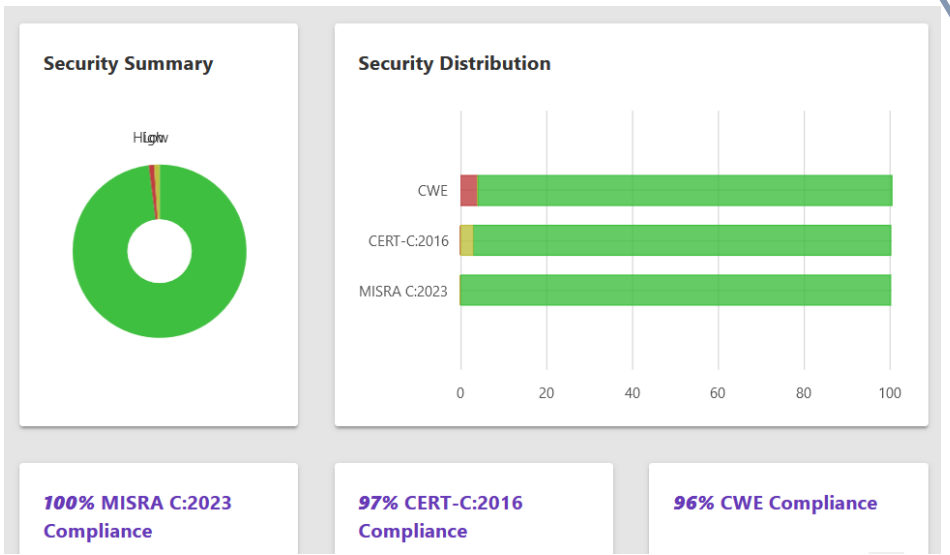
規格準拠の管理

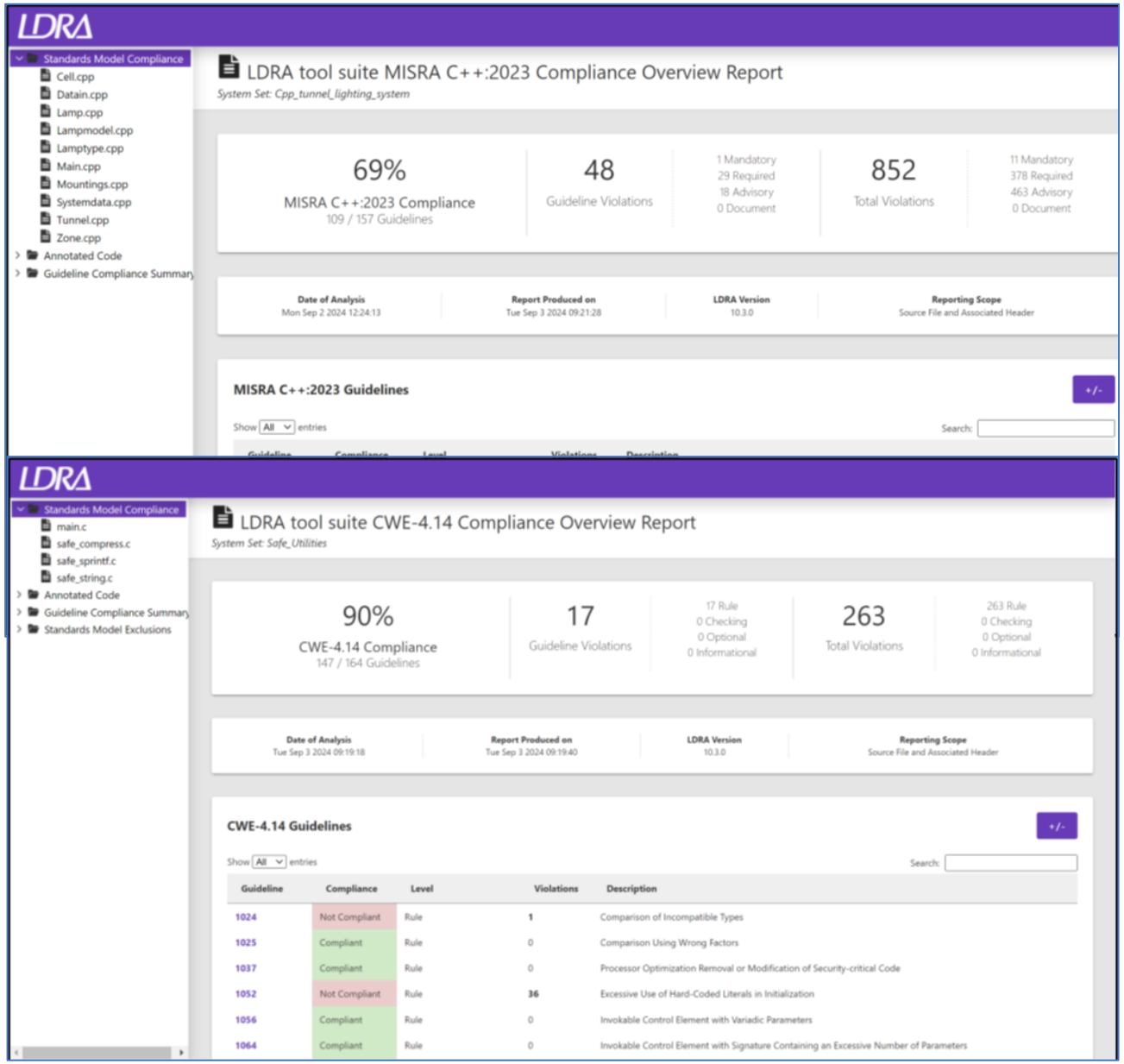
Certificate Of Quality

ISO 26262 &
ISO/SAE 21434
conformance

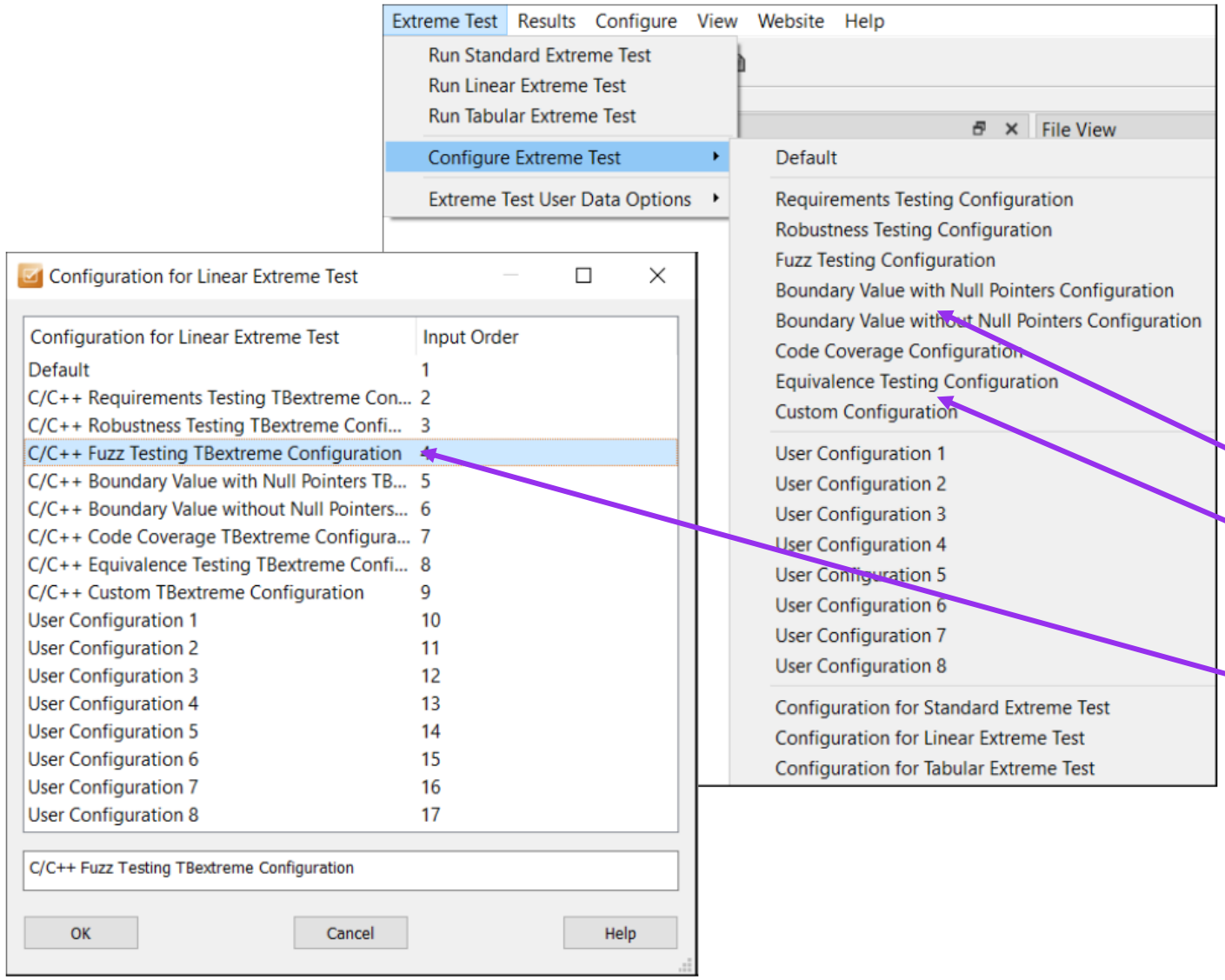
コーディング規約

- 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





テストケース自動生成

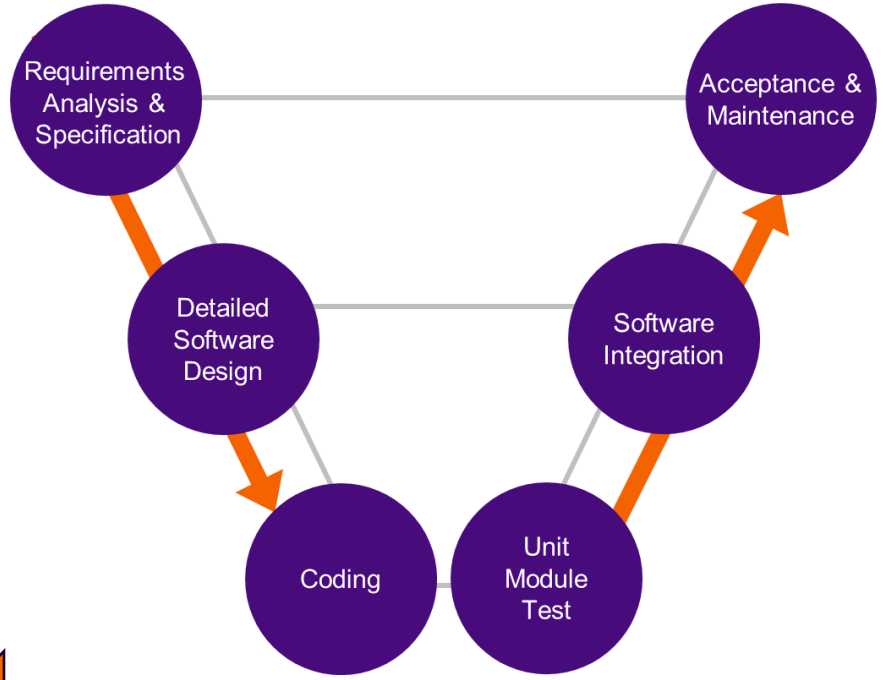


- 単体テストケース自動生成
- 機能安全規格の堅牢性試験
  - 境界値
  - 同値分割
- セキュアコーディングの
  ファジングテスト

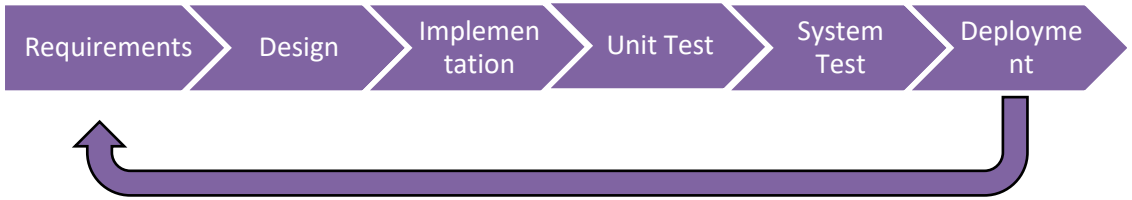
V字モデル、DevSecOps、検証作業の基本原則は同じ

開発

システム統合とテスト



シフトレフト



手動テスト

- 反復プロセスに親和的
- 速いサイクルタイム：実装 > ビルド > テスト > 反復
- 開発者のスキルセット
- 低いエントリポイント
- 使い慣れたツール

自動テスト

- コーディング規約準拠
- SAST（静的アプリケーションセキュリティテスト）
- 可読性、保守性、テスト容易性
- 安全性・セキュリティ主導のユニットテスト
- 構造カバレッジ解析

https://ldra.com/capabilities/devsecops/

LDRA

DevSecOps

顧客事例：航空・宇宙・防衛業界

DevSecOps 顧客事例

- 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

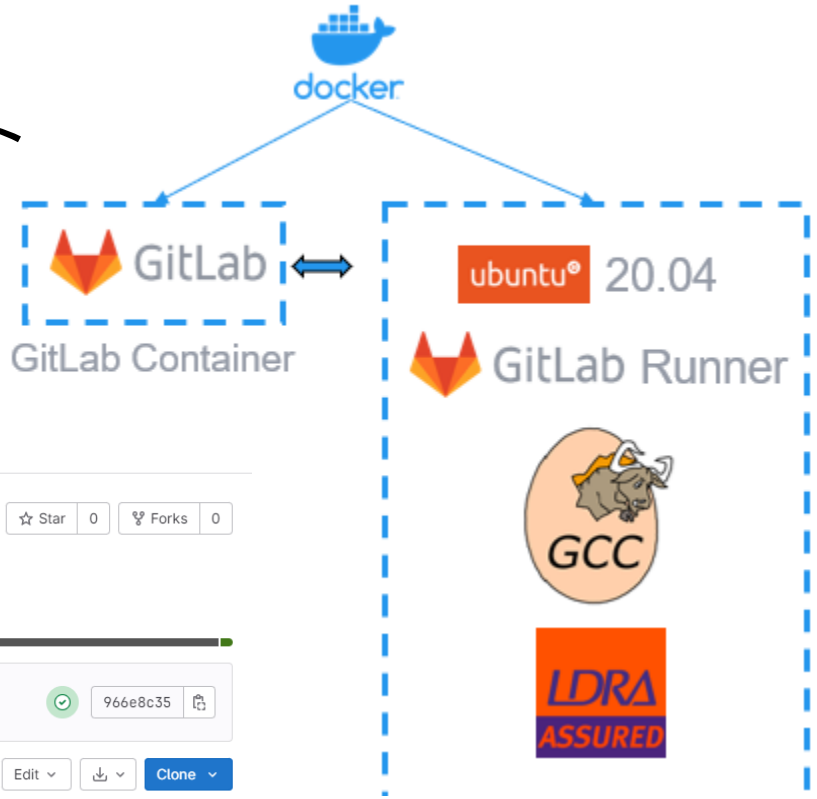
- 顧客事例 2:
  - 課題: 大規模なコードベースで静的解析、動的解析に長時間費やしていた
  - 解決策: 複数のコンテナを並列実行し解析時間を最大6倍速めることに成功

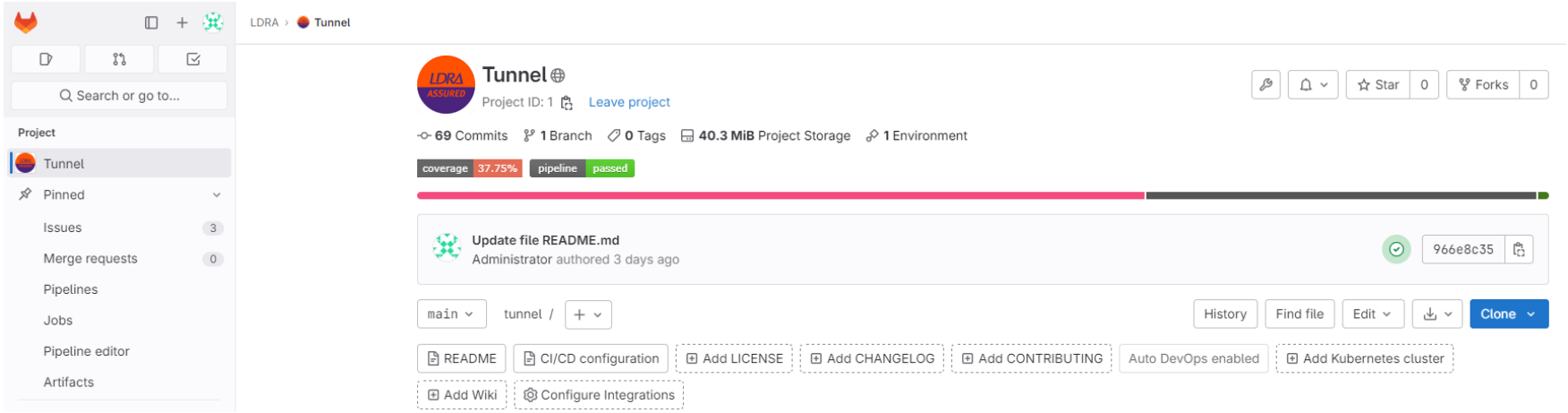
Domain	CPPs	CI/CD (no container)	Server tbwrkfls(8x licenses)	Localized tbwrkfls(8x licenses)	Status
App1	64	3h:37m	36m	24m	All pass
App2	30	1h:19m	13m	9m	All pass
CPI1	126	6h:59m	1h:19m	49m	All pass
CPI2	44	1h:28m	16m	11m	All pass
CPIX	61	2h:28m	37m	20m	All pass

WEBINAR ON-DEMAND DevSecOps for Military Applications – Best Practices for Development of Critical Software

顧客事例 2：GitLab、継続的インテグレーション

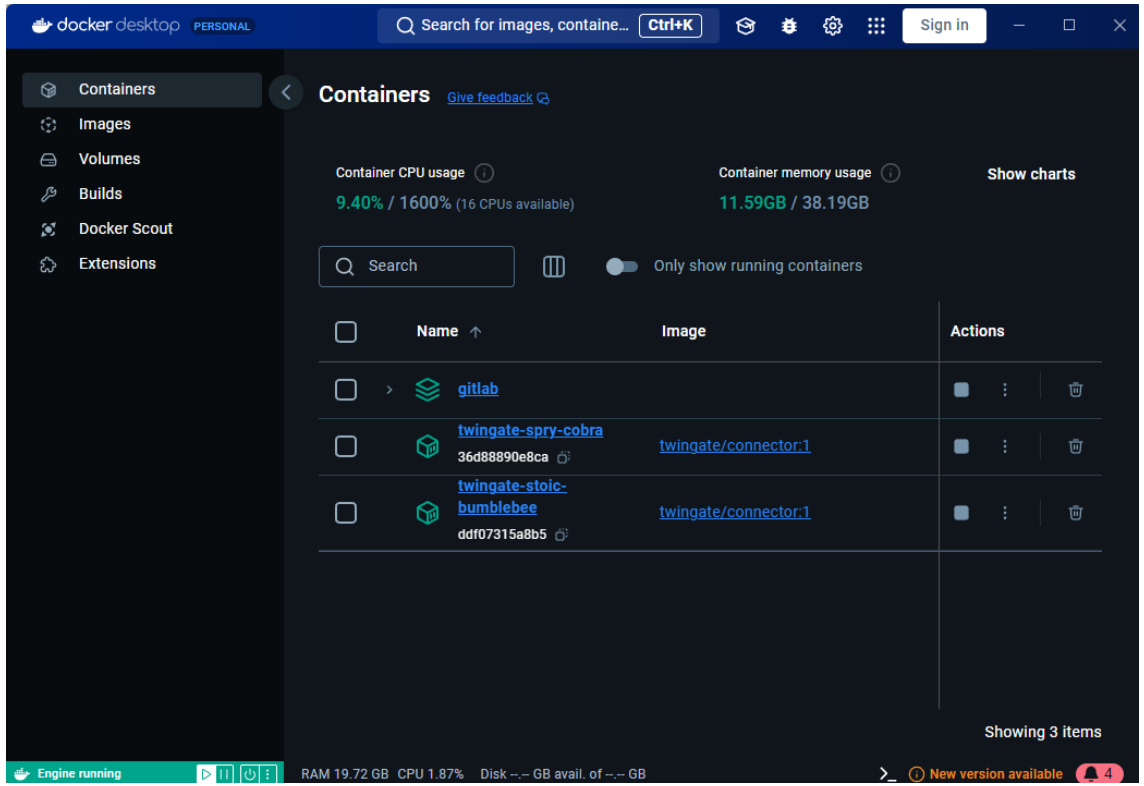
- GitLab を採用
  - 静的テストと単体テストを並列実行
  - 10,000 以上の TCF（テストケースファイル）で回帰テスト
  - チェックインから結果までのシームレスなワークフロー





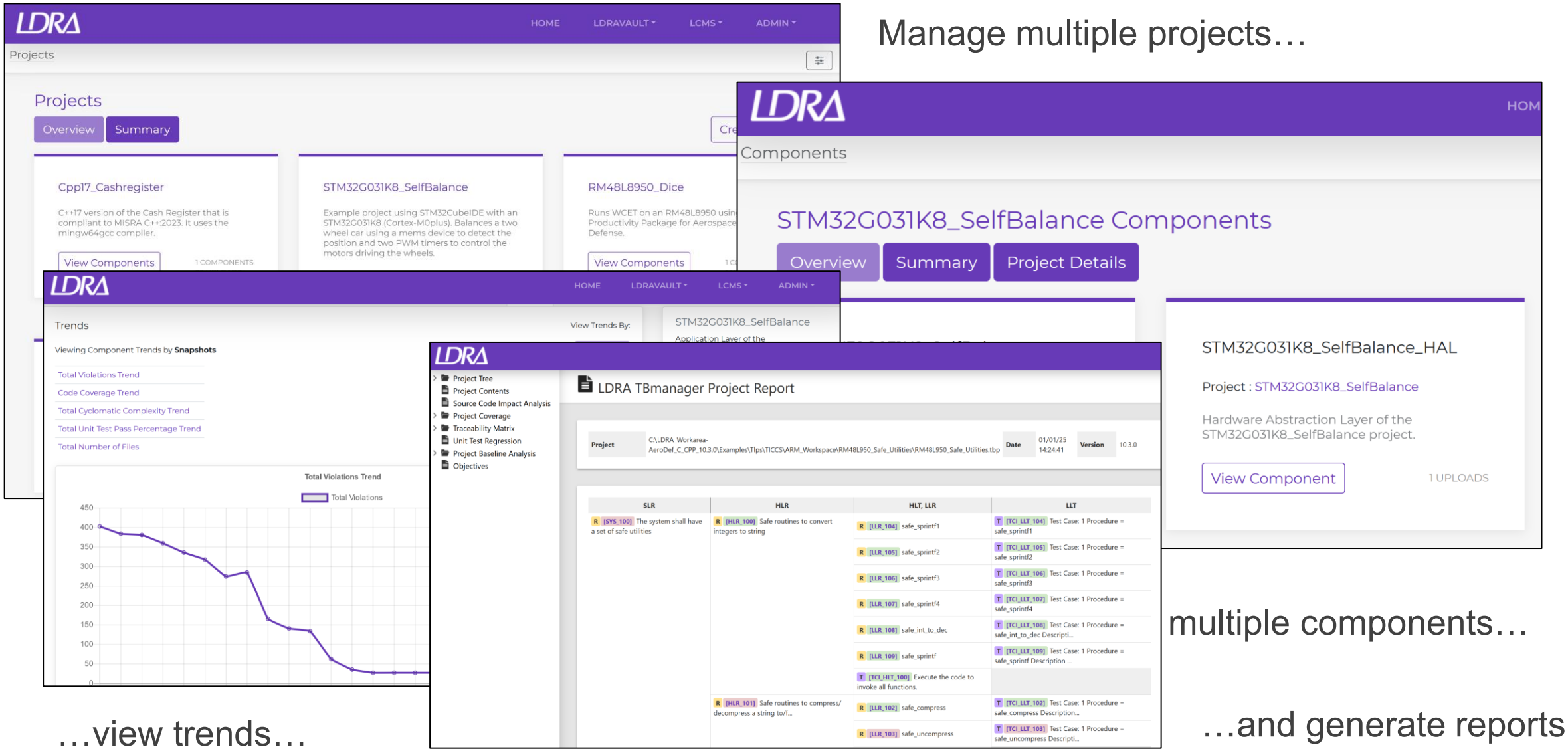
顧客事例 2:コンテナ化

- 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ルメント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



Manage multiple projects…

multiple components…

…view trends…

…and generate reports

顧客事例 2: 認証支援システムに統合

- 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

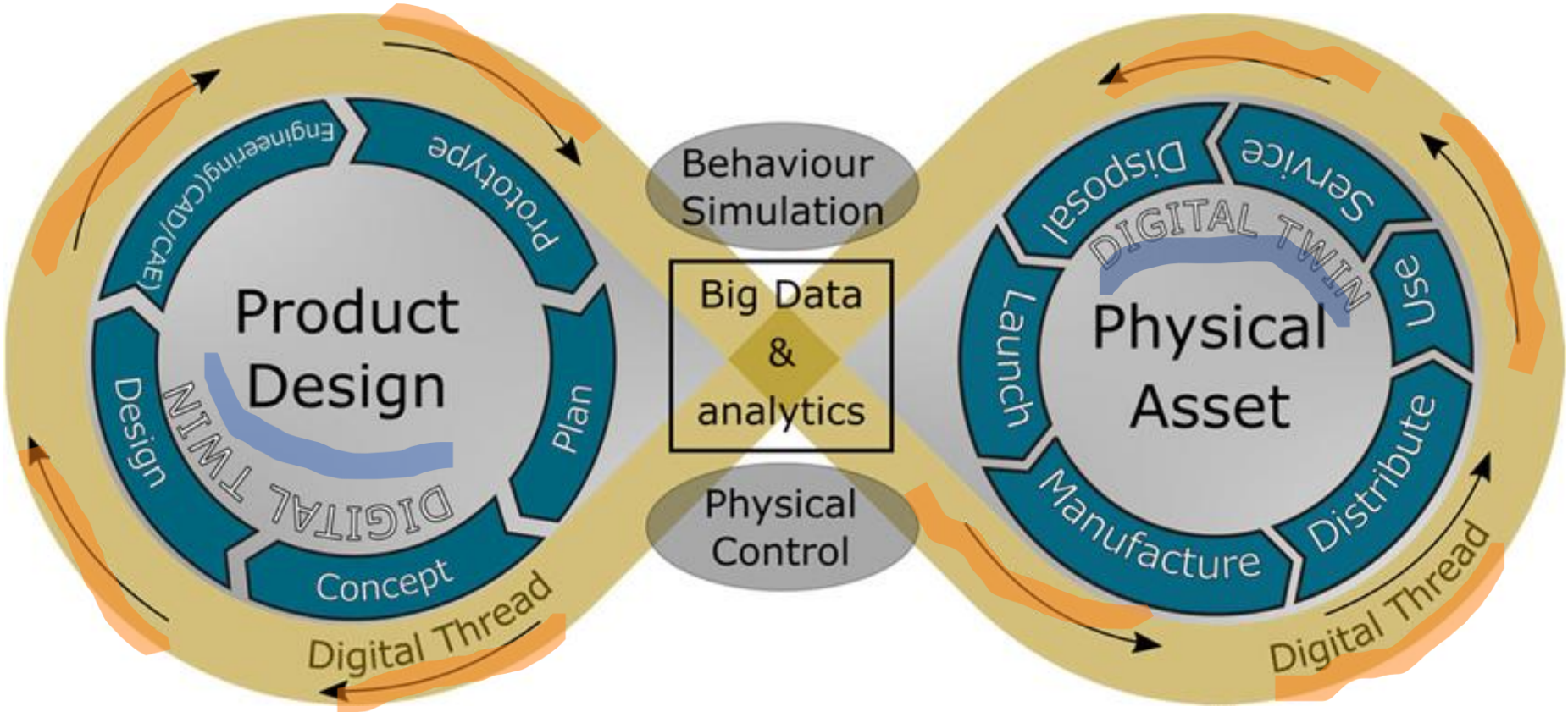
LDRA

DevSecOps

デジタルスレッド、デジタルツイン

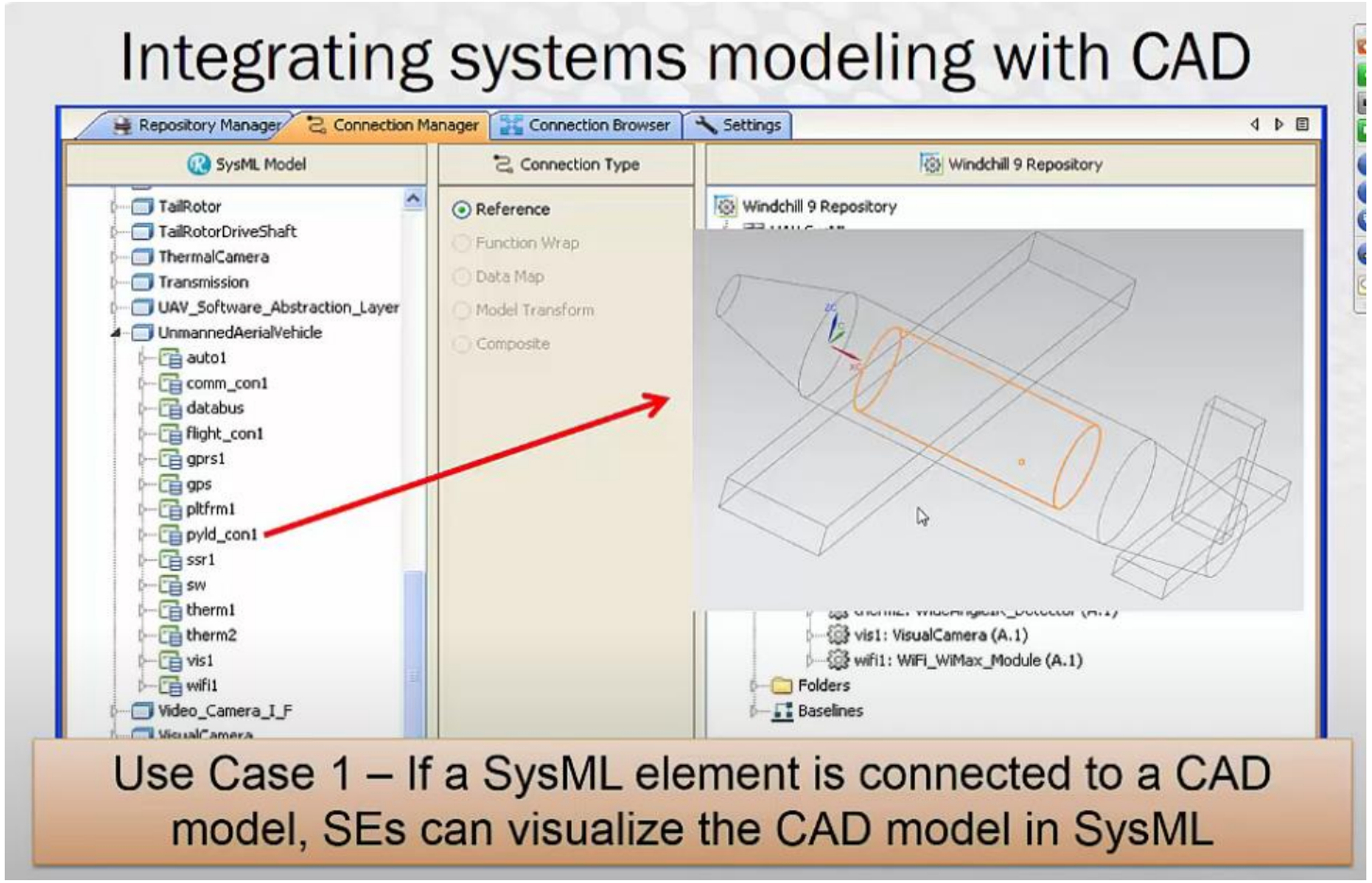
デジタルスレッド、デジタルツイン

- 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



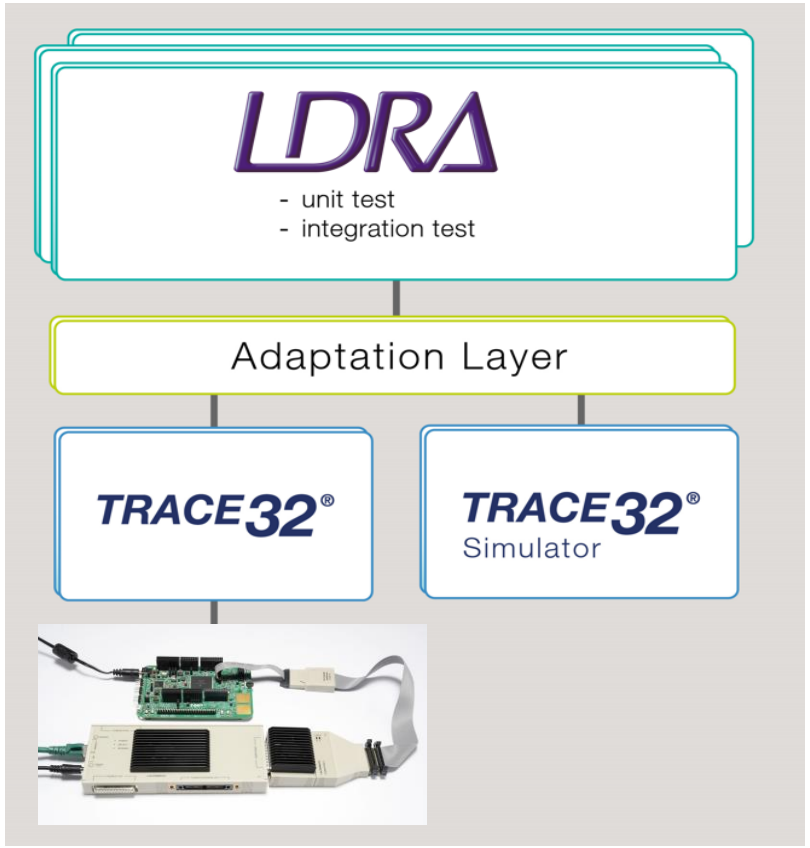
顧客事例 3：MBSE

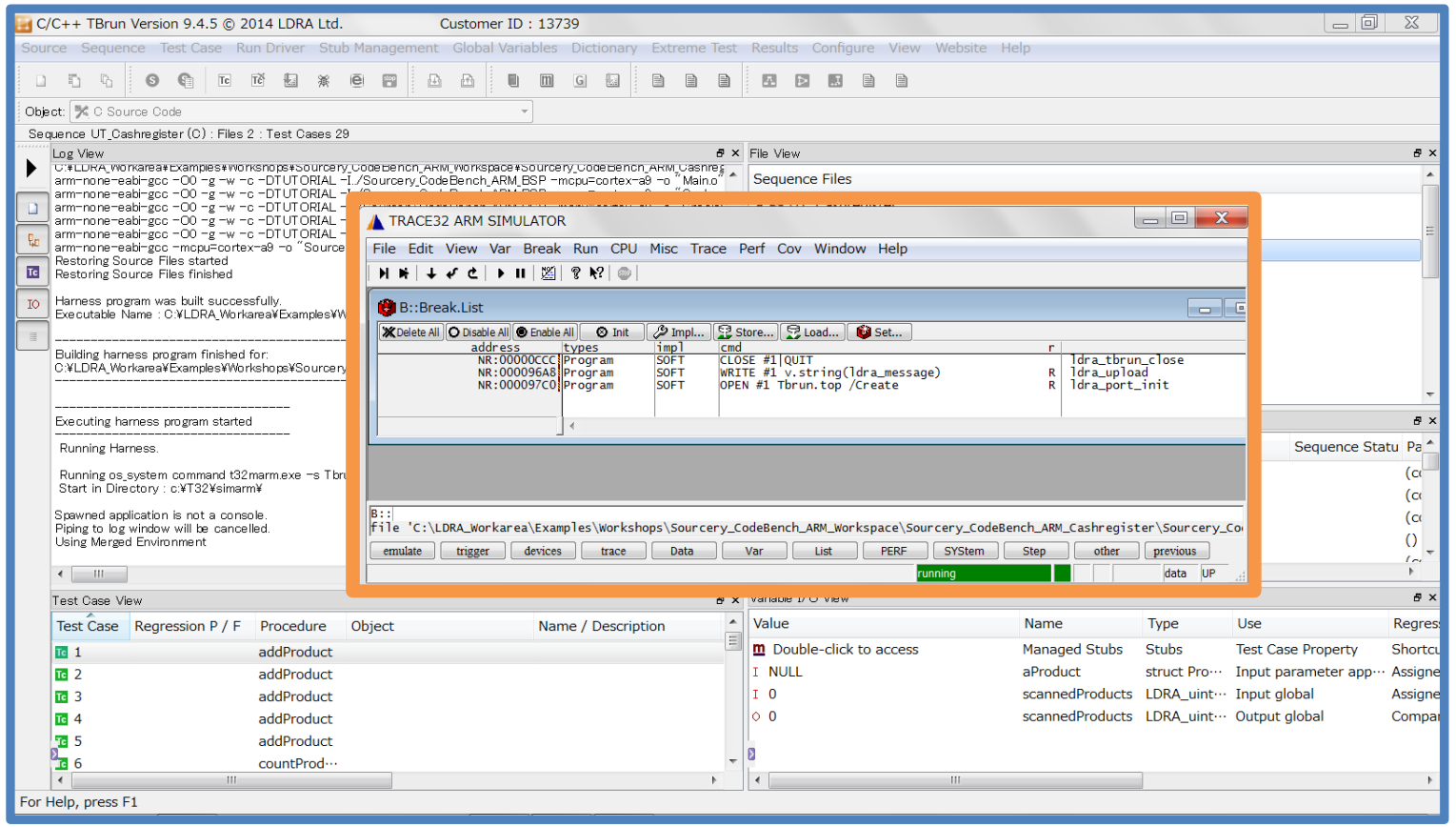
- デジタルスレッド：MBSEでCAD、モデル、コードをつなぐ
- デジタルツイン：モデルシミュレータ、インストラクションセットシミュレータ



デジタルツイン：シミュレータ、実機レベルで検証

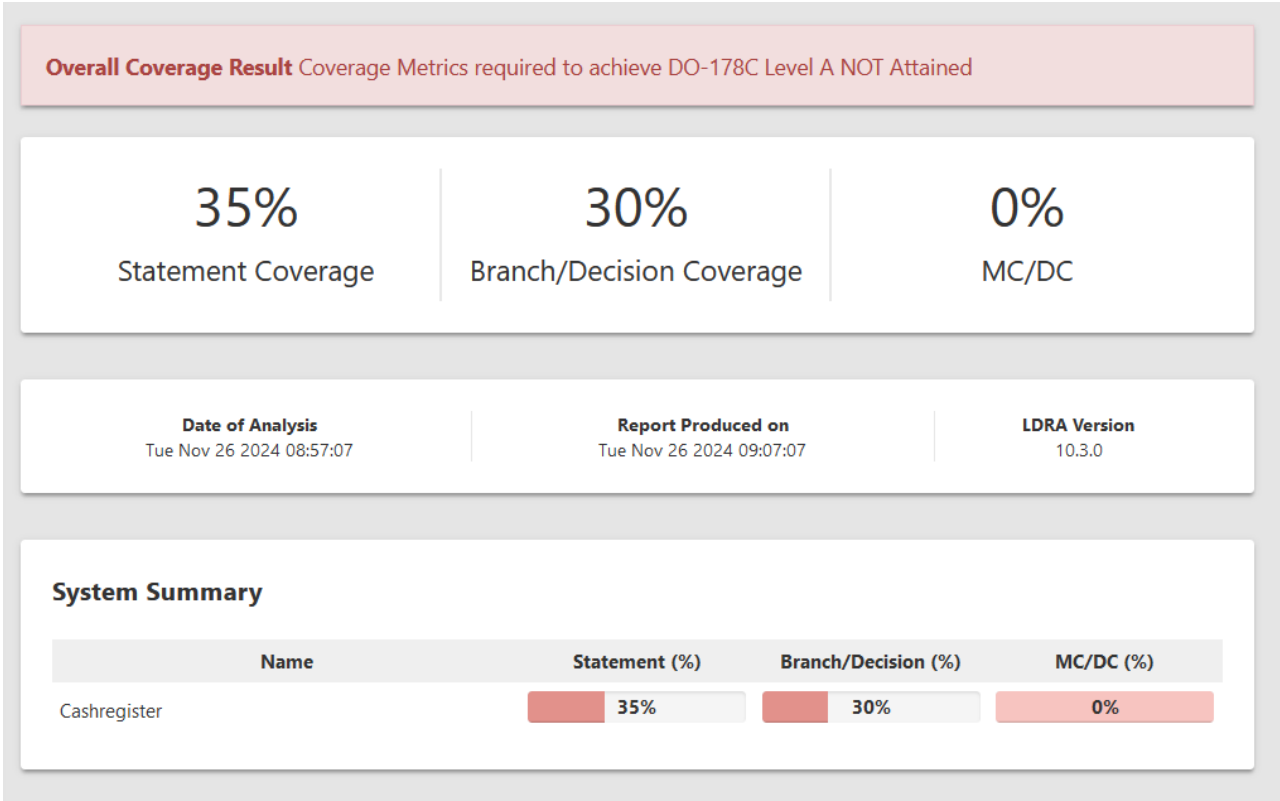
- あらゆるコンパイラ、デバッガ、実行環境での検証作業をサポート



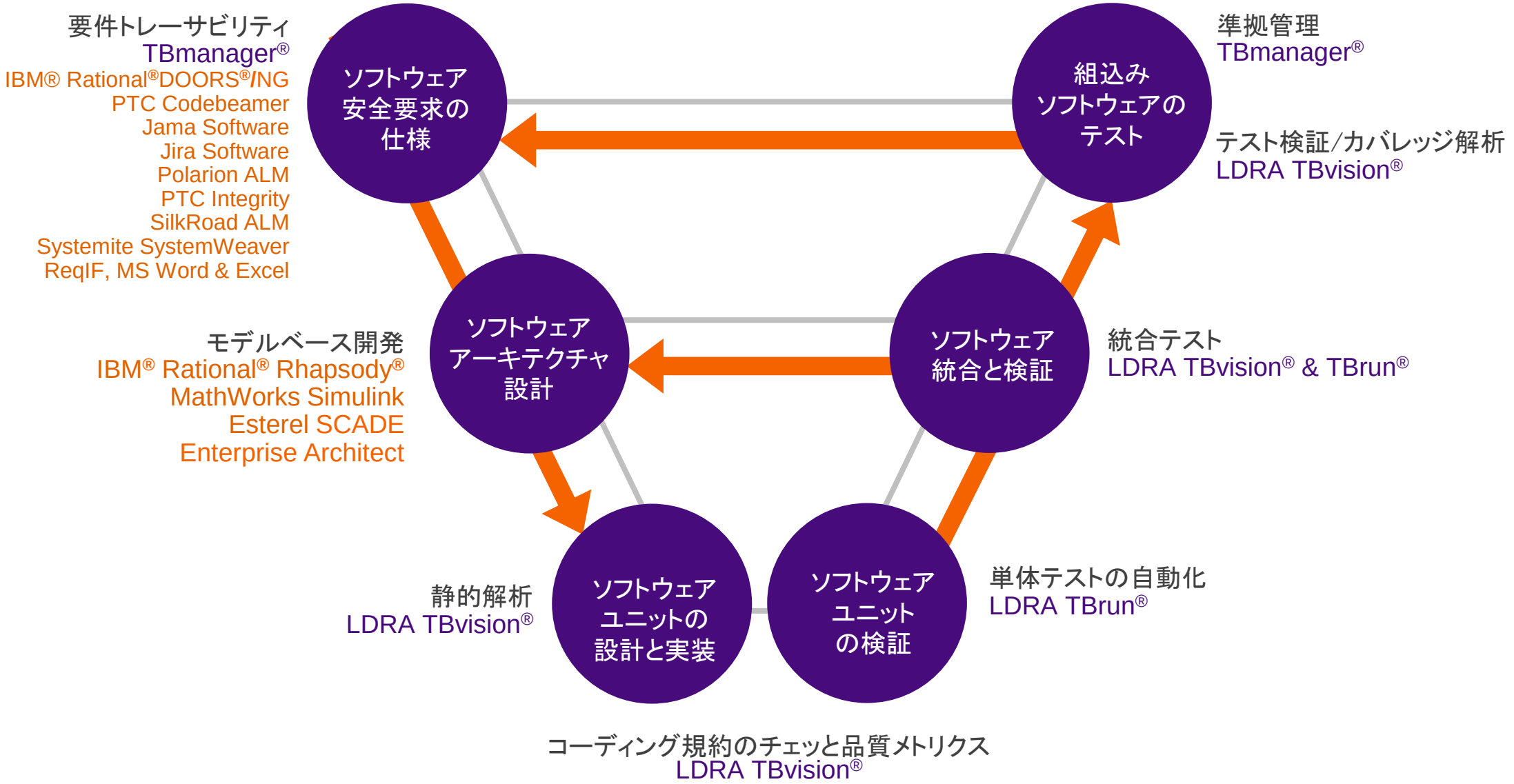


要件ベーステストとカバレッジ解析

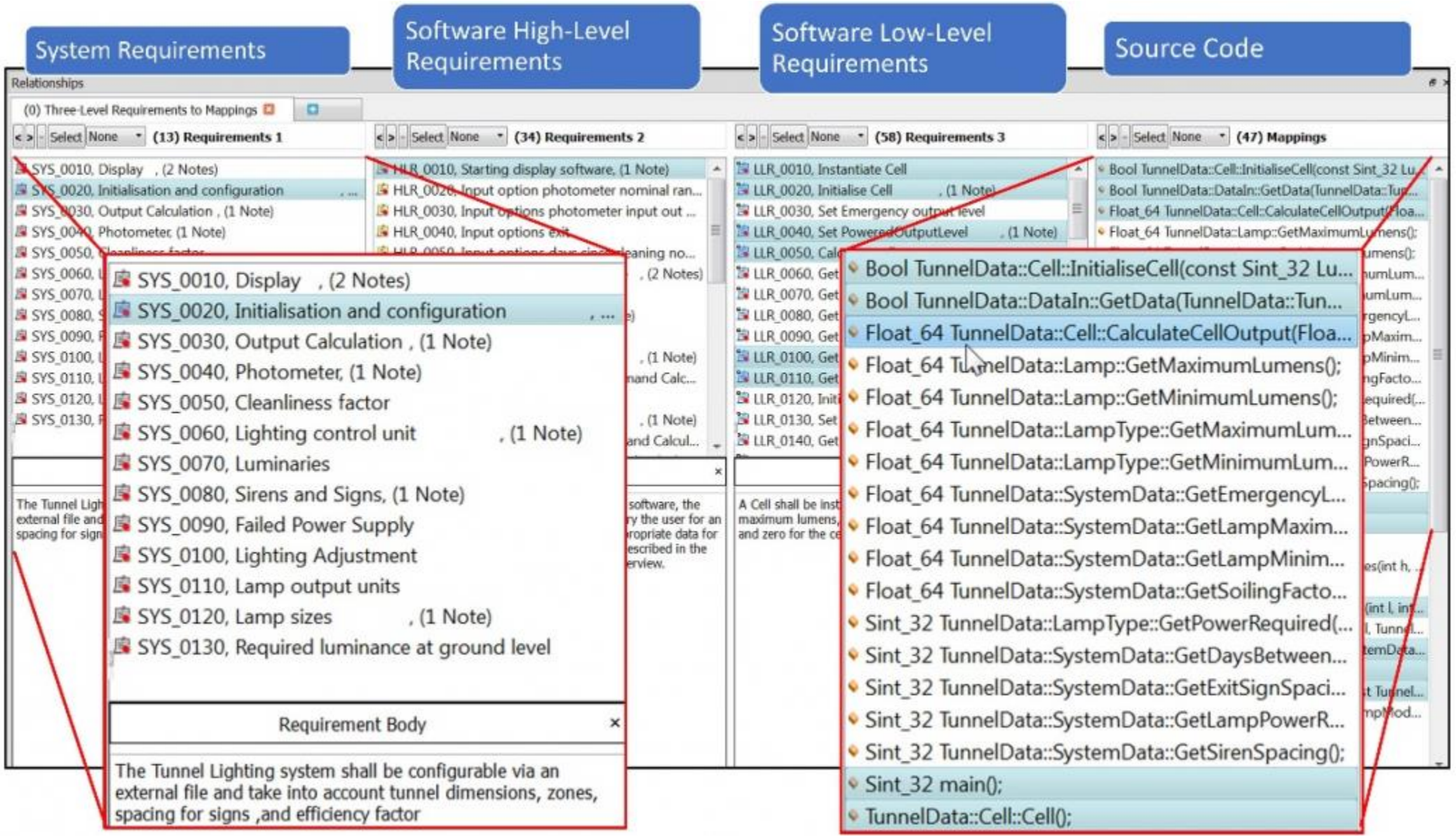
デジタルツインにおいても、要件ベーステストの十分性は構造化カバレッジやデータフローカバレッジ解析で評価され、デジタルスレッドの一部となる



V字モデルとデジタルスレッド

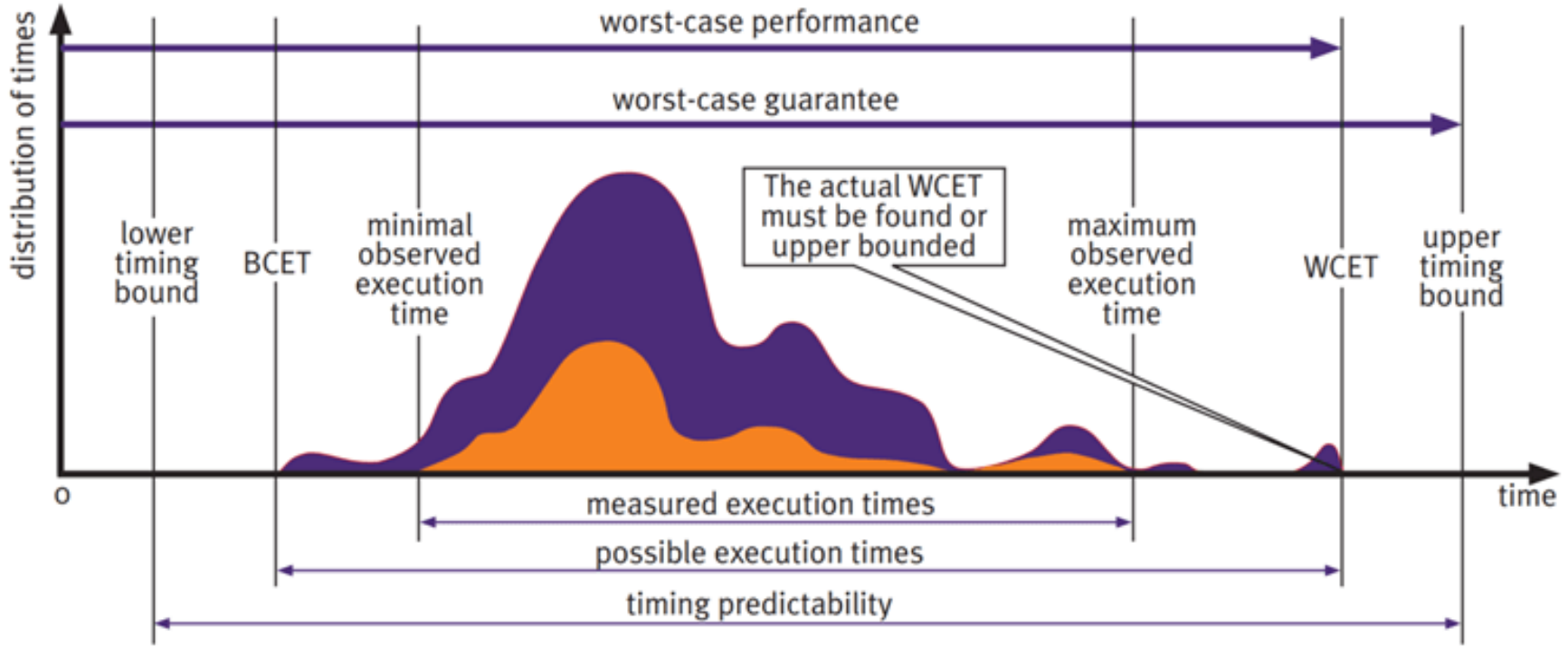


TBmanager 要件～テストを双方向にトレース



デジタルツインでタイミング解析

- 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

まとめ

機能安全ツールの価値提案

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）

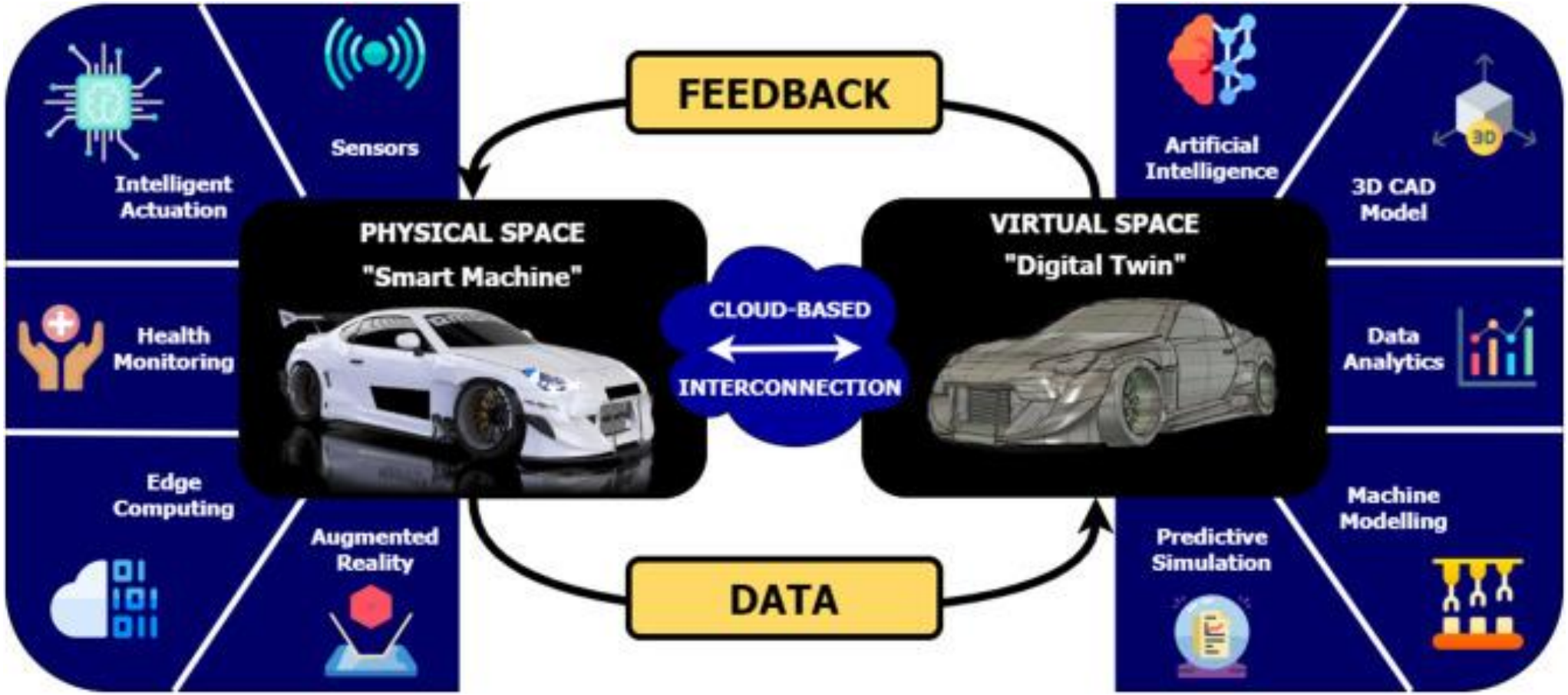
「ロケットが爆発したのにエンジニアたちが歓声を上げているというのは不思議に思うでしょう。なぜ歓声をあげるかというと、ロケットの爆発は本当の失敗ではなく期待以上の成果が得られたからなんです。
確かにロケットはとても高価ですが、ロケットよりも高価なもの、それは時間です。スペースX社は大量のデータを集めるのを非常に得意としています。ロケットには多くの機器を載せており、そこから常にデータが送られてきます。独自に開発したAIツールなどを駆使して、すべてのデータを分析。迅速に問題を修正し、成功に導いています。
つまり、大爆発したとしても、その過程で大事なことを学び、すぐに修正して次に進むことができるため、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

出典：NHK フロンティア 熱狂ムーンラッシュ なぜ月へ挑むのか
https://www.nhk.jp/p/frontiers/ts/PM34JL2L14/episode/te/JZ9X9GZV61/

SDV、DevSecOps



航空業界のベストプラクティスと DevSecOps

- MISRA、CERT、CWE 等コーディング規約
- 複雑性のメトリクスの測定
- テイント解析、データ結合/制御結合の解析
- 単体テスト、自動生成、カバレッジ解析
- 双方向の要件トレーサビリティ
- 静的～動的解析を統合して自動実行まで効率化

本講演は LDRA社 Jay Thomas 氏の講演を基に最新情報を追加しました

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。

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

FUJI SETSUBI

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/