安全性

組込みシステムの安全には、考慮すべき多くの側面があります。安全なシステム開発への総合的なアプローチでは、サイバーセキュリティ、人間工学、システムの完全性、環境要因、その他多くの要素を十分に考慮し、結果として得られる製品に関連するリスクが許容範囲内であることを保証します。

セーフティバイデザイン、機能安全(FuSa)、および SOTIF(意図した機能の安全性)は、ソフトウェア開発者がシステムの安全に対して行う主な貢献を表します。

セーフティバイデザイン(Safety by Design)とは何ですか?

セーフティバイデザインとは、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アプローチです。このアプローチには、安全要件を設計仕様に組み込むこと、徹底したリスクアセスメントを実施すること、冗長性とフェイルセーフメカニズムを実装すること、関連する安全規約を遵守すること、が含まれます。

安全に関する懸念に最初から対処することで、セーフティバイデザインは信頼性を高め、運用上のリスクを軽減し、システムまたは製品のライフサイクル全体にわたって安全パフォーマンスを最適化することを目指します。

航空電子機器の DO-178C 規格は、その後に続いた機能安全規格(つまり、IEC 61508 およびその派生規格)とは異なる、セーフティバイデザインのアプローチについて説明しています。

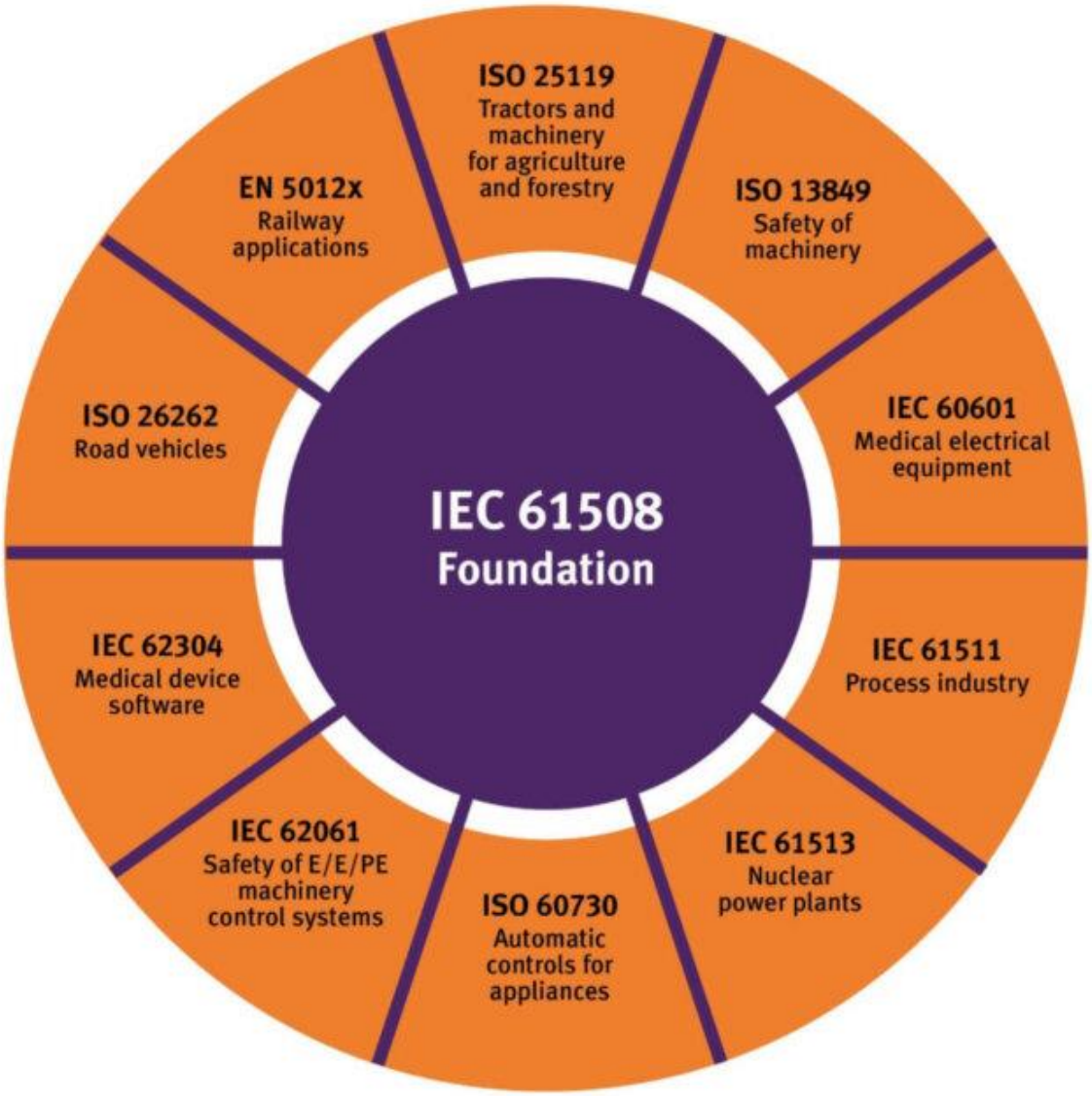
機能安全(FuSa)とは何ですか?

国際電気標準会議(IEC)は、この問題に対して次のような定義をしています。

「機能安全は、デバイスまたはシステムのリスクレベルを低減することを目指すものである。[機能安全は]危害をもたらす可能性のある潜在的に危険な状態を特定し、インシデントの影響を回避または軽減する是正措置を自動的に使用可能にするものである。これは、危険なイベントに対応する自動安全装置に依存するシステムまたはデバイスの全体的な安全の一部である。機能安全は、潜在的に危険な状況に対応できる能動的システムに依存している。」

IEC 61508 とその派生規格

IEC 61508「電気・電子・プログラマブル電子安全関連系の機能安全」は、開発ライフサイクル全体にわたって安全を管理するための包括的なフレームワークを提供します。このようなシステムの開発全体に直接適用できることに加えて、これは基礎となる規格でもあります。つまり、さまざまな分野の固有の要件に対応するために、多くの業界固有の規格がこの規格から派生しています。これらのいくつかを以下に示します。



セーフティバイデザインと機能安全の違いは何ですか?

IEC の機能安全の定義は、セーフティバイデザインと比較して、重点が微妙に異なります。

セーフティバイデザインでは、開発者はシステムの重要性に応じた要求を満たすようにシステムを設計・開発する必要があります。一方、機能安全規格では、開発者は開発された

システムがシステムの重要性に比例したレベルの完全性を備えていることを実証する必要があります。

セーフティバイデザインによるシステムは機能安全規格に準拠している可能性が高いですが、セーフティバイデザインでなくてもそのような規格に準拠することは可能です。

意図した機能の安全性(SOTIF)とは何ですか?

意図した機能の安全性(SOTIF：Safety Of The Intended Functionality)は、システムの障害や誤動作に関連する危険ではなく、システムの設計または環境が意図しない結果をもたらす状況から生じる潜在的な危険に対処します。

このような危険の例としては、予測できない要素との相互作用や予期しないユーザーアクションなどが挙げられます。SOTIF は、設計および開発段階でこれらのリスクを特定して軽減し、機能の正確性だけでなく全体的な安全パフォーマンスを向上させることを目指しています。

意図した機能の安全性と機能安全の違いは何ですか?

機能安全(FuSa)は、主に、障害や故障が発生した場合でも、システム内の安全関連機能が正しく動作することを保証することに重点を置いています。意図した機能の安全性(SOTIF)は、システムの意図した機能が正しく実行されているにもかかわらず危険な状況につながる場合に発生するリスクに対処します。したがって、FuSa と SOTIF は補完的です。

コーディング規約とは何ですか?

安全関連のコーディング規約は、セーフティクリティカルなシステムにおいて、セーフティバイデザイン、機能安全、または SOTIF 原則に従って開発されているかどうかに関係なく、重要な役割を果たします。これらの規約は、ソフトウェアコードの信頼性、正しい動作の信頼性、エラーや事故の発生可能性の低さを保証するためのガイドラインとベストプラクティスを提供します。これらの規約は、一般的なコーディングの欠陥を防ぎ、コード全体の品質を向上させ、国際規格の要件への準拠を保証するのに役立ち、ソフトウェア開発ライフサイクル全体を通じて安全なプラクティスを促進する上で重要な役割を果たします。

安全関連のコーディング規約の例は次のとおりです。

- MISRA C および MISRA C++

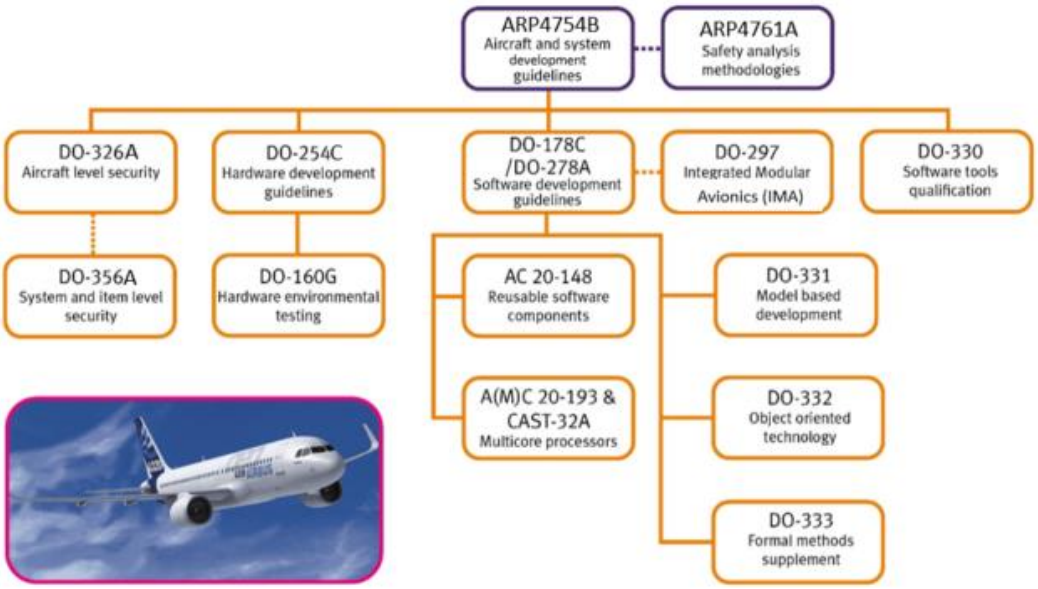
- BARR-C
- AUTOSAR C++（最近 MISRA C++ に統合されました）
- Joint Strike Fighter Air Vehicle C++（JSF 計画の F-35 戦闘機に採用）

航空宇宙および防衛における安全な組込みソフトウェア開発

航空宇宙および防衛（多くの場合、A&D と略）という用語は、民間部門と軍事部門の両方におけるさまざまな活動を網羅しています。この多様性にもかかわらず、安全にとって極めて重要なソフトウェアに対する共通のニーズは統一されています。

民間航空

民間航空当局は、航空電子工学ソフトウェアの安全規格を適用することで、組込みシステムのサイバーセキュリティの懸念に対処しています。この安全基準は、航空業界全体にわたって設計による安全性に対する総合的かつ一貫したアプローチを提供します。



DO−178C に関連する様々な規格

DO-178C/ED-12C は、すべての商用ソフトウェアベースの民間航空システムを承認するために認証機関が参照する主要な文書です。この文書はいくつかの文書によって補足されており、その多くは特定の開発アプローチや技術の使用をサポートしています。サポート文書の例は次のとおりです。

- DO-330/ED-215：ソフトウェアツールの認定を扱う独立した文書
- DO-331/ED-216：モデルベース開発を扱う DO-178C の補遺
- DO-332/ED-217：オブジェクト指向技術を扱う DO-178C の補遺
- DO-333/ED-218：形式手法を扱う DO-178C の補遺
- CAST-32A & A(M)C 20-193：マルチコアプロセッサの使用に関連する固有の認定課題に対処
- AC 20-148：複数のシステム間で再利用することを目的としたソフトウェアを扱う

防衛

防衛装備品に関する安全基準は国ごとに異なります。

- MIL-HDBK-516C ： 米国:「耐空証明基準」
- MIL-STD-883E ： 米国:「マイクロ回路の試験方法」
- DEF STAN 00-55 ： 英国:「防衛システムにおけるプログラマブル要素(PE)の安全性に関する要件」
- DEF STAN 00-56 ： 英国:「防衛システムの安全管理要件」

また、EU や NATO などによるものを含め、これらの国家規格を整合させるための取り組みもいくつかあります。多くの場合、防衛ソフトウェア開発者は民間規格にも準拠する必要があります。例としては、航空分野では DO-178C/ED-12C および DO-278A/ED-190D、地上車両分野では ISO 26262 などがあります。

自動車分野における安全な組込みソフトウェア開発

自動車業界では、特に車両の自律化やネットワーク化が進むにつれて、車両内の電子システムの信頼性の高い機能に重点を置いたガイドラインを通じてソフトウェアの安全性が確保されています。

機能安全（FuSa）

ISO 26262 – 自動車 ー 機能安全は、 IEC 61508 の派生規格です。自動車内の電気および電子システムの機能安全を実現するための包括的なフレームワークを提供します。

製品開発ライフサイクル全体をカバーし、ソフトウェア集約型システムの安全性に対処するために自動車業界で広く使用されています。

その他の機能安全規格は、ISO 26262 ガイダンスをさらに派生させて、より専門的なアドバイスを提供します。たとえば、ISO 25119 (トラクターおよび農業機械)は、修正および簡素化された ISO 26262 推奨事項(自動車の安全性)と ISO 13849 (機械類の安全性)を組み合わせています。

意図した機能の安全性(SOTIF)

ISO/PAS 21448 自動車 ー 意図した機能の安全性は、自動車システムの意図した機能の安全性(SOTIF)を扱っています。システムの適切な機能に関連する安全性の考慮事項に焦点を当てており、具体的には、システムが意図したとおりに動作しても危険な状況につながる可能性があるシナリオに焦点を当てています。これは、特に自動運転車に関連しています。

製造、プロセス、エネルギー分野における安全な組み込みソフトウェア開発

産業機器およびエネルギー部門は、幅広いドメインの商用および生産活動をカバーしているため、関連する規格が多数あります。これらは、一般的なものから非常に特殊なものまで多岐にわたります。次のものが含まれます。

- IEC 61508: 電気・電子・プログラマブル電子安全関連システムの機能安全
- ISO 13849 : 機械類の安全性 ー 制御システムの安全関連部
- IEC 62061 : 機械類の安全性 ー 安全関連の電気・電子・プログラマブル電子制御システムの機能安全
- IEC 60730 : 家庭用および類似の用途向けの自動制御
- IEC 61511 : 機能安全 ー プロセス産業分野の安全計装システム
- IEC 61513 : 原子力発電所 ー 安全上重要な計測制御 ー システムに関する一般的要件

これらの機能安全規格は、すべて IEC 61508 から派生しているため、多くの共通点があります。たとえば、それらはすべて、安全ライフサイクル、リスク軽減、共通原因故障を考慮する必要性を強調しています。

医療機器分野における安全な組込みソフトウェア開発

電子医療機器のソフトウェアの安全規格には、主にリスク管理に関する ISO 14971 や、汎用コンピューティングプラットフォームで動作する医療ソフトウェア製品の安全とセキュリティに重点を置いた IEC 82304-1 などがあります。

IEC 62304 ー 医療機器ーソフトウェアライフサイクルプロセスは、組込みソフトウェア開発者にとって最も重要な文書です。他の多くの安全に特化した機能安全規格と同様に、これは IEC 61508 から派生したものです。

FDA および欧州当局は、これらの規格がそれぞれの規制に準拠していることを示すものとして認識しています。たとえば、IEC 62304 および ISO 14971 は、FDA のガイダンス文書で頻繁に参照されており、欧州医療機器規則 European Medical Device Regulation (MDR) に基づいて整合されています。

鉄道（GTS）分野における安全な組込みソフトウェア開発

鉄道および GTS 業界では、クローズドな有線の分離ネットワークからオープンな相互接続ネットワークへの進化により、新たな脅威が生まれています。EN 50128 および EN 5012x シリーズ、その機能安全モデル、概念、およびそれが記述するリスク評価プロセスは、IEC 61508 規格に基づくか、そこから派生したもので、鉄道固有の状況に合わせて調整されています。さらに、鉄道安全指令および相互運用性指令では、欧州鉄道ネットワーク全体の安全性と相互運用性に関する必須要件を規定しています。

欧州以外では、ソフトウェアの鉄道安全規格として、EN 50128 と同等の国際規格である IEC 62279 が含まれます。米国では、連邦鉄道局(FRA)と米国公共交通協会(APTA)が鉄道安全に関するガイドラインと規格を提供しています。

宇宙分野における安全な組込みソフトウェア開発

宇宙分野に関わるリスクが高いことから、ソフトウェアの安全基準は非常に重要です。

組込みソフトウェア開発に関する NASA のガイダンス文書は、規格（STD）と手順要件（NPR：NASA Procedural Requirements）の 2 つのグループに分かれています。NPR はプロセスと管理の側面が NASA の目標と一致することを保証し、STD はそれらの目標を達成するための技術およびエンジニアリングの規格を設定します。NP 7150.2D ソフトウ

ェアエンジニアリング要件は、セーフティクリティカルな組込みソフトウェアの開発に最も直接関連する文書です。

さらに、欧州宇宙標準協会(ECSS)が発行する ECSS シリーズの規格は、要件定義、設計、製造、検証、妥当性確認、保守など、宇宙ソフトウェアエンジニアリングのあらゆる側面をカバーしています。これらの規格は、ミッションや安全を脅かす可能性のある故障を防止するために、厳格なテスト、妥当性確認、リスク管理を重視しており、特に最も重要なシステムのソースコードからオブジェクトコードへのトレーサビリティを重視しています。

安全な組込みソフトウェアと LDRA ツール

さまざまな分野で提供されるアドバイスは、内容、詳細、成熟度が異なりますが、共通のテーマも多数あります。これには次のものが含まれます。

- 最初に、または各イテレーションの前に機能要件とセキュリティ要件を確立する
- 開発のすべての段階にまで双方向に要件をトレースする
- 安全な言語サブセットを使用する
- 安全を重視したプロセス規格を遵守する
- テストを自動化し、コードカバレッジ解析を実行する
- 早めに頻繁にテストする
- 認定された検証ツールを使用する

LDRA ツールスイートを含む LDRA の検証支援ツールは、関連する安全規格への準拠、およびコードの設計と検証に対する要件のトレーサビリティを求める開発チームが直面するオーバーヘッドを軽減するのに役立ちます。

最初に要件を確立する

要件が不明確であったり、文書化されていなかったりすると、コミュニケーションエラーにつながり、やり直し、変更、バグ修正、セキュリティの脆弱性が生じる可能性があります。プロジェクト開発をスムーズに進めるには、すべてのチームメンバーがすべての製品コンポーネントと開発プロセスについて共通の理解を持つ必要があります。この連携には、明確に定義された機能要件とセキュリティ要件が不可欠です。

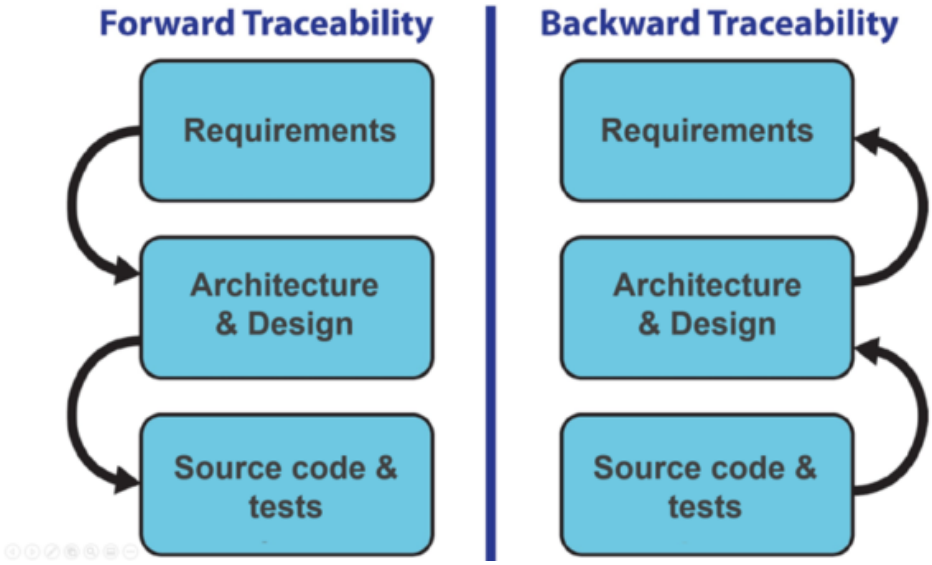
V 字モデルでの開発者にとって、これらの要件は通常、完全なシステムを定義しますが、DevOps のような反復プロセスでは、開発者は1回の繰返しの概要だけを記述することも

あります。いずれにしても、ソフトウェアは概念実証のために「知的モデリングクレイ」として使用できますが、正式な開発プロセスでは、明確に定義された要件と、それらを満たすように開発されたコードが含まれている必要があります。

双方向のトレーサビリティを提供する

IEEE 標準ソフトウェアエンジニアリング用語集では、トレーサビリティを「開発プロセスの2つ以上の成果物、特に相互に先行-後続または主従関係を持つ成果物間で関係を確立できる度合い」と定義しています。双方向トレーサビリティとは、トレーサビリティパスが前方と後方の両方で維持されることを意味します。

自動化によって、変化するプロジェクト環境でトレーサビリティを維持することがはるかに容易になります。



双方向トレーサビリティ

前方トレーサビリティは、実装とテストを含む開発プロセスの各段階ですべての要件が反映されていることを示します。要件の変更や失敗したテストケースの影響は、影響分析を適用することで評価でき、その後対処できます。その結果として得られた実装を再テストして、双方向トレーサビリティの原則が継続的に遵守されているという証拠を提示できます。

同様に重要なのは、指定された要件をまったく満たさないコードを強調表示する後方トレーサビリティです。見落とし、ロジックの誤り、過度の機能追加、悪意のあるバックドアメソッドの挿入はすべて、セキュリティの脆弱性やエラーを引き起こす可能性があります。

LDRA ツールスイートの TBmanager コンポーネントによって実現される自動トレーサビリティは、特に要件が変更された場合に進捗状況を追跡する上で非常に役立ちます。

安全な言語のサブセットを使用する

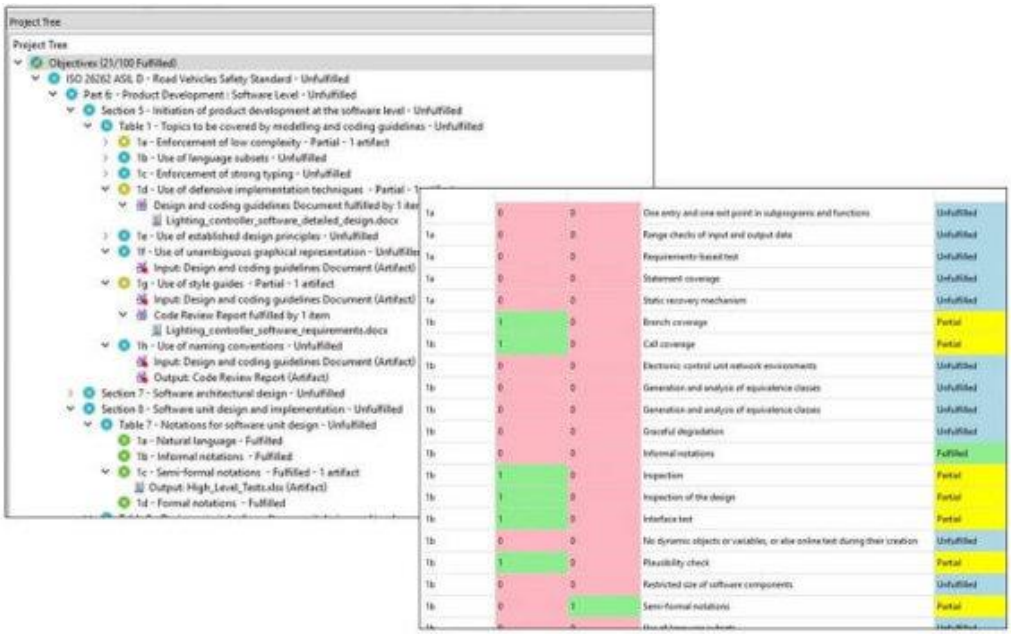
C や C++での開発では、ソフトウェアの欠陥の約 80%が言語の約 20%の誤った使用に起因していることが、長年の研究でわかっています。これに対処するために、開発者は問題のある構造を禁止することで安全性とセキュリティの両方を向上させる言語サブセット(コーディング規約)を使用できます。

MISRA C や BARR-C などの規約を熱心に適用すると、より安全で高品質なコードが実現される可能性が高くなります。大規模なコードベースにこのような規約を手動で適用するのは現実的ではありません。ソースコードの「検査」を自動化するには、静的解析ツールが必要です。

この点で、LDRA ツールスイートの TBvision や LDRArules が提供する静的分析が役立ちます。

安全を重視したプロセス規約を遵守する

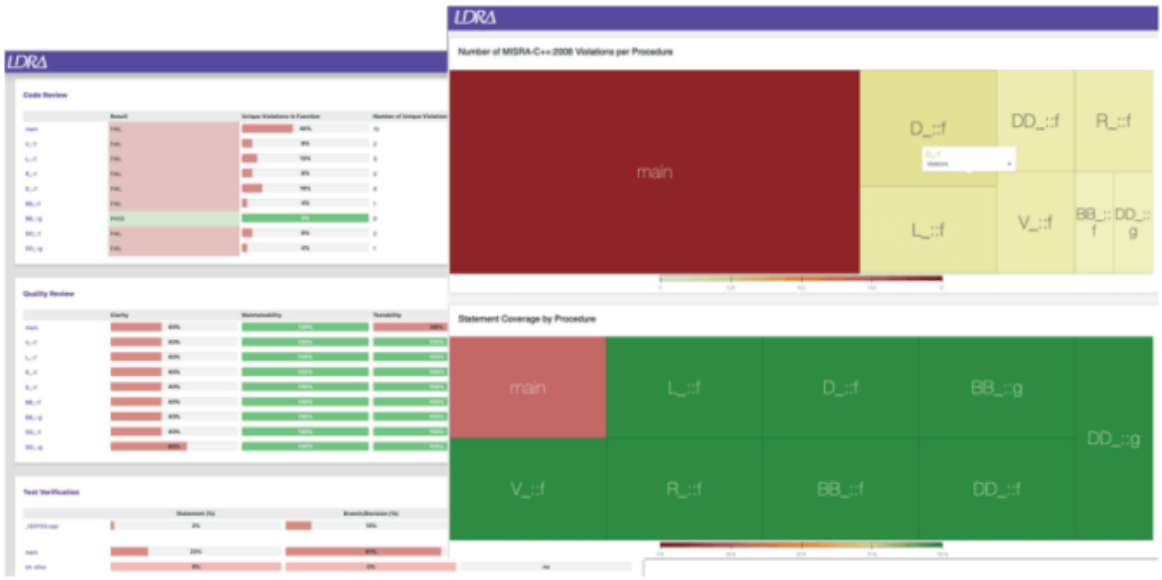
安全性を重視した規約に従うことは、多くの場合賢明であり、必須です。LDRA ツールスイートの TBmanager コンポーネントは、必要に応じて、セキュリティクリティカルな安全と機能安全を同時に実現する開発者ワークフローを統合することもできます。



TBmanager

LDRAVault は、TBmanager を補完するものです。これは、複数のプロジェクトにわたる膨大な量のデータを処理できる、Web ベースのエンタープライズレベルのアプリケーション認

証集約ツールです。LDRAツールスイートのアドオンである LDRAVault は、認証目的の証拠となる成果物として使用するのに適したレポートと結果を自動的かつ安全に集約する Web インターフェイスを提供します。例としては、コードレビュー、コードカバレッジ解析、ユニットテストの結果などがあります。



LDRAVault

テストプロセスの自動化

静的解析とは異なり、動的解析ではテスト対象のコードベースの一部またはすべてが実行されます。

LDRA ツールスイートは、ソフトウェアの品質を向上させ、安全関連規格への準拠を保証するために設計されたさまざまな動的テスト機能を提供します。主要な動的テスト機能には、次のものが含まれますが、これらに限定されません。

- ユニット、システム、および統合テスト ： ホスト、ターゲット、シミュレータでの単体テストと統合テストを容易にするためのテストハーネスとドライバプログラムを生成
- コードカバレッジ解析 ： コードの未テスト部分を特定し、要件のトレーサビリティを実証するための詳細なコードカバレッジメトリックを提供
- データ結合と制御結合の解析 ： 依存関係を最適化することを目的に、あるモジュールの別のモジュールへの依存度を解析
- 実行時間分析(WCET) ：干渉解析などマルチコアプロセッサのサポートを含む

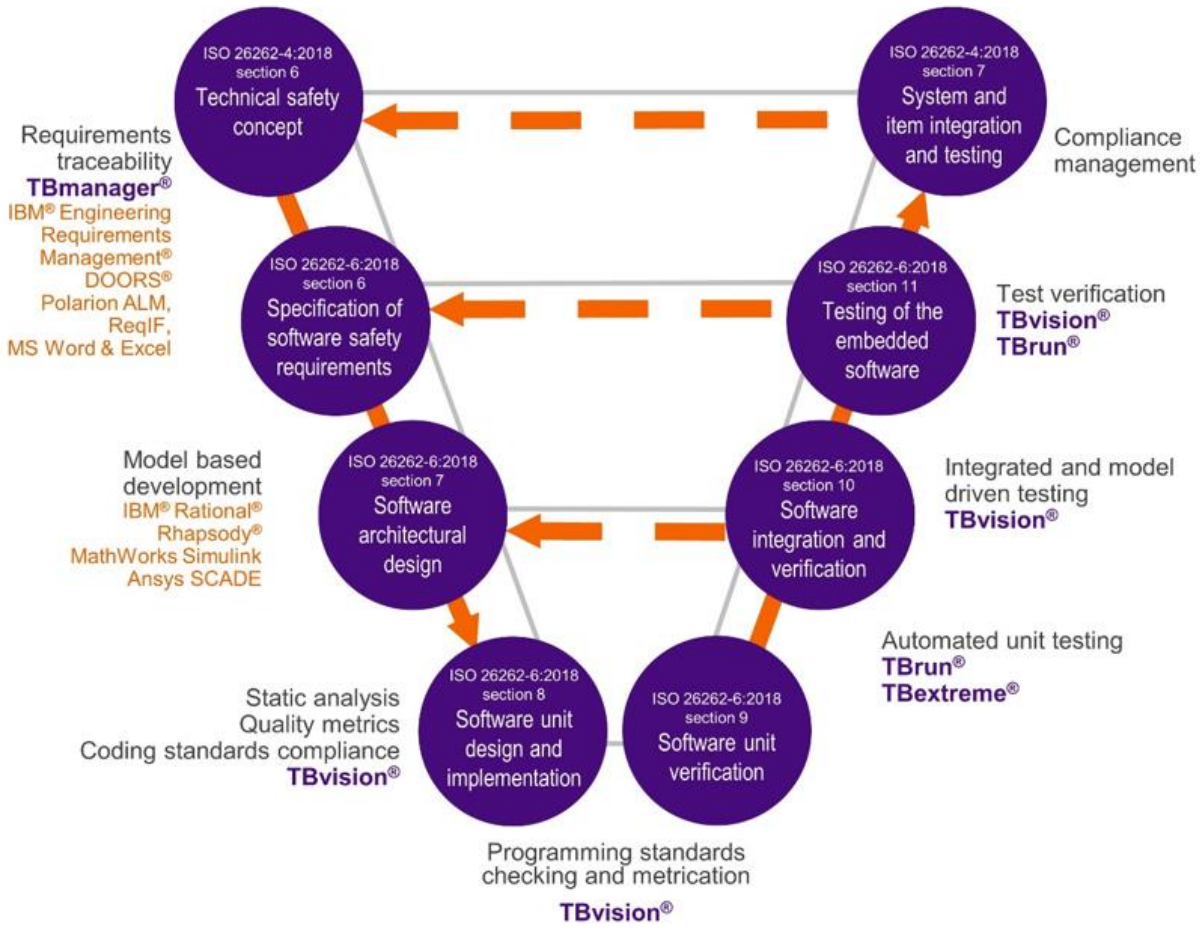
- ソースコードからオブジェクトコードへのトレーサビリティ ： コンパイラが開発者の意図を忠実に再現する実行可能オブジェクトコードを生成したことを確認する手段

これらの機能は、開発者が開発プロセスの早い段階で問題を特定して修正するのに役立ち、ソフトウェアの全体的な信頼性とセキュリティを向上させます。

早めに頻繁にテストする

ここで説明するすべての安全関連のツール、テスト、および技法は、様々な開発ライフサイクルモデルで活用できます。ISO 26262 の V 字モデル図では、これらが標準で説明されているプロセスとほぼ類似しており、補完的であることを示しています。各技法は、できるだけ早く活用されるように設計されています。これは、早期に特定された問題が最も簡単に修正できるためです。

同じ原則は他のプロセスモデルにも同様に当てはまり、自動車業界では DevOps、スパイラル、アジャイル、スクラムなどの反復プロセスモデルが特に人気があります。



ISO 26262 V 字モデルでの各種テスト手法とツール

認定された検証ツールを使用する

「ツール認定」とは、ツールエラーがシステムの安全性に影響を与えるリスクが許容できるほど低いことを保証するように設計されたプロセスを表す一般的な用語です。エラーが少ないか、またはエラーが安全性に影響を与えないことです。ほとんどの機能安全規格では、ツールの用途とツールがデプロイされる環境を考慮してツール認定を達成する必要性について言及しています。

航空電子工学の世界（特に DO-178C と DO-278A）では、文書 DO-330 に詳細に記述されているツール認定プロセスが、ほとんどのプロジェクトに適用できます。実際には、これは通常、ツールプロバイダーのサポートによって実現されます。たとえば、TQSP は、LDRA ツールスイートで使用できるオプションです 。各 TQSP モジュールは、プロジェクト固有の環境で検証ツールとして使用するために LDRA ツールスイートを認定するプロセスを簡素化するための成果物とガイダンスを提供します。

ISO 26262 を除き、IEC 61508 とその派生規格では、極めて重要なシステムについてはツール認定を取得する必要があると言及されているだけです。そのため、多くの開発者は、自動車および航空電子工学分野以外では、ガイダンスとして ISO 26262 または DO-330 を参照しています。この課題を認識し、DO-330 は、分野に関係なく適切なガイダンスを提供できるように意図的に設計されています。

SGS TÜV SAAR と TÜV SÜD は、どちらも独立した認証機関であり、LDRA の開発とテストの実践を評価し、LDRA ツールスイートが安全関連のソフトウェア開発で使用するのに適格であると確認する証明書を発行しています。LDRA ツールに対する TÜV 認証は、IEC 61508 およびその派生規格に準拠したプロジェクトへの適合性を十分に保証するものです。

この記事は https://ldra.com/safety/ をベースに日本語化したものです。

富士設備工業株式会社 電子機器事業部
https://www.fuji-setsu.co.jp