自動車用機能安全規格ISO26262の紹介

ISO26262 Automotive Functional Safety Standard

茂野 一彦*
Kazuhiko Shigeno

自動車用機能安全規格ISO26262が2011年11月に制定された。これを受けて、自動車業界ではこの規格の適用を標準化する動きがある。

こうした状況の中で、自動車メーカのみならず自動車メーカに部品を納入しているサプライヤの中には、ISO26262の要件を満たすため、いち早く体制を整え、規則を整備し、規格に基づくプロセスを構築して第三者機関による認証を取得したところもある。

当社も近いうちにこういった事業者を顧客とする部門において、規格の要件を満たす対応を迫られる可能性が高い。そこで、そもそもISO26262とはどのような規格かを紹介する。

November 2011 International Organization for Standardization standardized the ISO 26262 Automotive Functional Safety Standard, Since then there has been a global movement to adopt it as a standard of the automotive industry. In this industry, not only the car manufacturers but also some of car component suppliers have already organized their systems, establishing necessary rules and precisely fitting process to the standard, and then gaining ISO 26262 certification by third parties.

Also our company, being involved in the above industry, may have to meet the requirements of ISO 26262 in the near future. So now we are having an idea of the standard and will explain here the outline of it.

1. まえがき

現在、自動車業界では、電気／電子部品とそこに搭載されるMPU（マイコン）のソフトウェア開発において、自動車用機能安全規格ISO26262の適用が標準化されようとしている。

当社の車事業を扱う部門においても、今後、ISO26262に対応する車載機器の制御ソフトウェア開発を請け負う場合、顧客からISO26262に基づく要件を満たすよう求められる可能性が高い。

そこで、そもそもISO26262とはどのような規格であるのか。そのポイント（２章～７章）と、当社の業務に関わりの深いソフトウェア開発において何が求められるか（８章～11章）を紹介する。

2. 機能安全

機能安全とは、安全機能や安全対策によって、許容できないリスクから免れるための技術の総称である。機能安全（functional safety）の「機能」とは、制御対象やコントローラを監視する安全装置の役割のことを指す。通常、安全装置にはコンピュータが使われ、コントローラに故障などが発生した場合は、このコンピュータが制御対象を停止したり、ユーザに警告を出したりする。安全装置によって実現されているこうした安全性のことを、「機能安全」と呼ぶ。機能安全とは、いわばコンピュータなどを使った安全装置による安全対策といえる。機能安全の例としてよく引用されるのは、鉄道の踏み切りである。踏み切りは、警報システムの機能によって鉄路を走る列車と鉄路に平面交差する道路を通る車両や人との衝突を防いでいる。

なお、安全性そのものは、こうした電子的な安全装置の付加によって担保するのではなく、危険そのものの設計上の除去や機械構造的なフェールセーフ機構などによって担保するのが一般的である。これを「本質安全」と呼ぶ。本質安全の例でよく引用されるのは、鉄道と道路の立体交差である。これは機械構造的に鉄路を走る列車と道路を通る車両や人とが衝突する危険そのものを取り除いている。

＊関西事業部 姫電プロジェクト室

機能安全の考え方は、石油化学プラント、原子力発電、産業機械、航空機、鉄道、自動車、医療機器など、多岐にわたるクリティカルな（事故が発生すると人命が脅かされたり、環境に重大な影響が及ぶような）システムに適用されている。

ここでは、自動車の機能安全について取り上げる。

3. 機能安全規格制定の経緯

機能安全の規格化の動きは、1990年代から始まった。その背景には産業のあらゆる分野で電子化が進み、ソフトウェア（特に組み込みソフトウェア）で動作する機器が非常に増えてきたことがある。即ち「複雑で大規模なソフトウェアを含むシステムにリスク・ゼロはあり得ない」という共通認識が安全やソフトウェアの専門家の間に広がり、電子系に関する安全基準を各社が独自に策定・運用する方法ではもはや限界があるとされた。こうした動きを受けて、プロセス産業における電気／電子／ソフトウェアの機能安全に関わる国際規格「IEC 61508」が2000年に制定された。

IEC61508では、システムを構成するリレーやPLC（Programmable logic controller）などの安全装置（リスク低減手段）が、その安全機能を果たす確からしさ（信頼度）という確率論的な指標を安全度水準SIL（Safety Integrity Level）として定義している。即ち、システムを構成する安全にかかわる部品の故障率を低くすれば、それによって構成されるシステム全体も故障率が低いという定量的な確率論に基づく規格である。

しかし、SILの定義に確率的な要素が入るのは、適切でないという考え方がある。なぜ不適切なのか。それは機能安全規格がハードウェアだけでなくソフトウェアも扱っているからである。ハードウェアであれば、初期故障や磨耗故障以外の偶発故障はほぼランダムに発生するため、設計エラーを別とすれば、確率論的な扱いは非常にうまく当てはまる。これに対し、ソフトウェアの不具合（バグ）は、その発生を確率的には扱いにくい。例えば、ソフトウェアの設計にバグが混入していた場合、その発生経路や条件が整えば、不具合の現象は100％現れる[(1)]。

そこで、IEC61508の確率論に偏重した機能安全の問題点を改めたのが、2011年11月に制定された自動車用安全規格ISO26262である。

ISO26262では、確率論に基づく定量的なハザード分析をハードウェアに限定し、システム全体としては対象製品の使用状況、使用方法の定性的な分析結果に基づくハザードを特定する。そして、このハザードを評価する指標（これをASIL（Automotive Safety Integrity Level）という）を定め、ASILのレベルに応じた方策によるシステムの開発を行って、結果的に事故発生のリスクを低減している。

4. ISO26262とは

「ISO26262」とは、車載電子システム向けの機能安全規格である。

車載電子システム、つまり車両に搭載する電気／電子機器とコンピュータ（ソフトウェアを含む）がISO26262の直接的な対象である。対象とする車両は重量3.5ton以下の乗用車で、２輪車やトラック、障害者用などの特殊用途向け車両は含まれていない。

ISO26262の目的は「安全」の確保であり、その実現のためには、ISO26262の直接的な対象となる電子システムに限らず、システムを構成する他の要素も含めた安全性の考慮が必要となる。その上で、ISO26262の適用箇所を明確にし、システム全体における車載電子システムの安全性を体系的に確保する必要がある。

ISO26262の全体構成を図１に示す。

太枠で囲んだ矩形がPart（部）を示しており、図からわかるように、この規格はPart １～Part10で示された計10個のPartから構成されている。その中にある細枠の矩形は各Partにおける規定の節（項目）であり、個々のプロセスに対応する。

Part １は規格で使われる用語の定義を行っている用語集である。

Part ２の機能安全の管理では、安全関連系の開発に関わる組織や人員が満たすべき要件を規定する。

Part ３のコンセプトフェーズではASIL決定の手続きとしてのアイテム[(注1)]定義、ハザード分析とリスクアセスメント、ASILとともに設定される安全目標から機能安全コンセプト[(注2)]を作成する手続きを行う。

Part ４のシステムレベルの製品開発では、Part ３で得られた機能安全コンセプトを技術安全コンセプト[(注3)]に詳細化し、システム設計に反映する。そして、ハードウェアとソフトウェアに該当する安全要件を仕様化する工程（図１の4- 5、4- 6、4- 7）とハードウェア／ソフトウェアの開発成果物を統合して評価（検証および妥当性確認）を行う工程（図１の4- 8～4-11）を策定する。

Part ５のハードウェアレベル製品開発では、確率論

（注１） ISO26262を適用する車両レベルの機能を実装するシステム。
（注２） 安全目標の達成に必要な機能安全要件を導き出し、それに基づいた要件をアイテムの暫定的なアーキテクチャの構想、および外部方策に適切に割り当てること。
（注３） アイテムレベルの機能安全要件をシステムレベルに詳細化した技術的な安全要件仕様。

凡例 2-5 2：Part 番号
5：Part 毎の節番号

図1 ISO26262の全体構成

に基づくハードウェアの偶発的故障発生確率(注4)の計算、技術安全コンセプトに沿って開発する安全メカニズム(注5)の実装、および、その結果として得られる安全目標を侵害する確率の定量的評価に基づいた設計と試験を行う。

Part 6のソフトウェアレベルにおける製品開発では、V字プロセスに従って、技術安全コンセプトに従ったソフトウェア安全要件の導出、ソフトウェアアーキテクチャ設計、ソフトウェア単体設計と詳細化を実施し、それを

基にコーディング（実装）を行った後は、ソフトウェア単体試験、ソフトウェア統合（モジュール結合）と試験、ソフトウェア安全要件の検証を行う。

Part 7は量産やサービス、市場の監視、さらには廃棄するまでの安全要件を規定する。

Part 8はサプライヤへの開発委託、支援系プロセス（安全要件の管理、構成管理、変更管理、検証、文書化）および、ソフトウェアツール認定、ソフトウェアコンポーネント(注6)認定、ハードウェアコンポーネント(注7)認定に関する要件を規定し、複数のプロセスに対して横断的に関与する。

Part 9はASILの取り扱いと技術的な分析手法の指針を規定し、Part 8と同様、複数のプロセスに対して横断的に関与する。

Part10は規格の本編であるPart 1～9で記載困難または書ききれなかった特定項目の解説および事例を示したガイドラインである。

次章以降で、ISO26262の体系に則り、ハザード分析とリスクアセスメントから機能安全のソフトウェア開発に至る流れを概説する。

（注4） ハードウェア部品の寿命中に確率分布に従い発生する、予期できない故障の発生確率。
（注5） 電気・電子ハードウェア、ソフトウェアやその他の技術によって、安全状態を確保するために障害を検出したり、故障を制御したり、または異常警告を発してリスクを回避したりする技術的解決策。
（注6） 一つまたは複数のソフトウェアユニット（ソフトウェア関数）。
（注7） 一つまたは複数のハードウェアユニット（ハードウェア部品）。

5. ハザード分析とリスクアセスメント

ISO26262では、ハザードを「アイテムの機能不全の振る舞いにより引き起こされる危害になりうる原因」、リスクを「危害が発生する確率とその危害の過酷度（Severity）との組み合わせ」と定義している。

安全は元から備わっているものではなく、目標を立てて達成するものであり、規格はその目標の指標を示すものである。ハザード分析とリスクアセスメントの目的は、不合理なリスクを避けるために、アイテムの機能不全（故障）が引き起こすハザードを特定および分離し、危険事象の防止または緩和に関連した安全目標を立てることにある(2)。

ハザード分析では、状況分析とハザード識別を実施することにより、危険事象につながる可能性のあるアイテムに潜在する危険につながる振る舞いを識別する。ハザード識別にはFMEA（Failure Mode and Effect Analysis）やHAZOP（HAZards and OPerability study）といった手法が使われる。

6. ASILの決定

ISO26262では、リスクの指標としてASILを使用する。

ASILは、リスクアセスメントの結果得られる3つの指標（S、E、C）から決定する。

S：過酷度（Severity）

これは、アイテム故障の振る舞いの結果によってドライバまたは他の交通関係者が受ける傷害の重さの見積もりである。危険事象のそれぞれに対し、表1にしたがってS0、S1、S2、S3の過酷度のクラスの1つを割り付ける。

この指標は、医学的なガイドラインなどをもって見積もられる。

表1　過酷度のクラス

クラス	S0	S1	S2	S3
内容	傷害なし	軽度および中程度の障害	重度および生命を脅かす障害（生存の可能性がある。）	生命を脅かす傷害（生存がはっきりしない）

E：ハザードの発生頻度（probability of Exposure）

これは、想定される運転状況の期間、もしくは、ある状況の発生頻度のどちらかの指標による見積もりである。危険事象のそれぞれに対し、表2にしたがってE0、E1、E2、E3、E4の発生頻度のクラスの1つを割り付ける。

この指標は、対象となる危険事象のシナリオ（危険事象から事故に至るまでの流れ）に基づき、道路環境、天候、車両周辺状況、時間帯、運転操作などから判断される。

表2　ハザードの発生頻度のクラス

クラス	E0	E 1	E2	E3	E4
内容	可能性なし	可能性が非常に低い	可能性が低い	可能性が中程度	可能性が高い

C：回避可能性（Controllability）

これは、ドライバまたは他の潜在的リスクのある人が、特定の危害を回避するために危険事象に対し十分に抑制することができる確率の見積もりである。危険事象のそれぞれに対し、表3にしたがってC0、C1、C2、C3の発生頻度の1つを割り付ける。

表3　回避可能性のクラス

クラス	C0	C1	C2	C3
内容	一般的に回避可能	容易に回避可能	通常は回避可能	回避困難または回避不可

この指標は、ドライバあるいは他の被害を被る可能性のある交通参加者が、発生している危険事象を制御でき、被害を回避することができる可能性の見積もり評価である。

こうして危険事象のそれぞれに割り付けたS、E、Cの各クラスから、表４にしたがってASILを決定する。

表４において、QM（Quality Management）は機能安全を適用しなくてもよい通常の品質管理である。安全ライフサイクルを実施する組織はISO／TS16949（自動車産業向け品質マネジメント規格）、ISO9001（品質マネジメント規格）または同等の規格に準拠した組織レベルの品質管理が必要であるとされる。

表４　ASILの決定表

過酷度のクラス	ハザードの発生頻度のクラス	回避可能性のクラス		
		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	ASIL A
	E4	QM	ASIL A	ASIL B
S2	E1	QM	QM	QM
	E2	QM	QM	ASIL A
	E3	QM	ASIL A	ASIL B
	E4	ASIL A	ASIL B	ASIL C
S3	E1	QM	QM	ASIL A
	E2	QM	ASIL A	ASIL B
	E3	ASIL A	ASIL B	ASIL C
	E4	ASIL B	ASIL C	ASIL D

7. 製品開発における機能安全要件の継承

本章では、アイテムの安全要件がどのように継承されるかを説明する。

図２に安全要件の継承フローを示す。

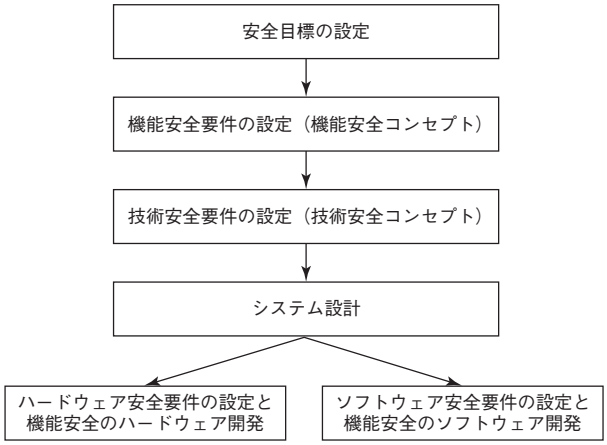


図２　安全要件の継承フロー

7.1　安全目標の設定

５章、６章で説明した、ISO26262を適用するアイテムに対するハザード（危険源）の分析とリスクアセスメントおよびASILの決定は、受入れ不可能なリスクを避け、アイテムの安全目標を決定するために実施する。安全目標とは、特定の運転状態において、危険事象につながるハザードに対処するための機能的な要求である。

安全目標にはハザード分析で評価された各々の危険事象と関連するハザードに対して決定したASILを割り付け、IDを付加する。

7.2　機能安全要件の設定（機能安全コンセプト）

安全目標に従うため、機能安全コンセプトにおいて、アイテムの機能安全要件を規定する。

機能安全コンセプトとは、安全目標の達成に必要な機能安全要件を導き出し、それに基づいた要件をアイテムの暫定的なアーキテクチャの構想、および外部方策（他のアイテムに対してリスクを軽減する解決策）に割り当てることである。

機能安全要件とは、安全目標に基づくアイテムの安全な振る舞いの仕様および実装に依存しない安全方策（有害な影響を軽減するためのアクティビティまたは解決策）である。

安全目標と機能安全要件の関係は図３に示すような階層構造となっている。

7.3　技術安全要件の設定（技術安全コンセプト）

機能安全要件をアイテムに組み込んで実現するためには、技術安全コンセプトによってアイテムレベルの機能安全要件をシステムレベルに必要な技術安全要件に詳細化しなければならない。

技術安全要件とは、機能安全要件を実装するためにシステムが備えるべき技術的な安全方策である。

技術安全コンセプトとは、技術安全要件をどのように実現すべきかを示す仕様である。

機能安全要件の詳細化は、アイテムを構成する要素ごとに技術安全要件を仕様として策定することで行う。

7.4　システム設計

システム設計では、技術安全要件を含んだシステム要求仕様の実装方法を設計する。ここでは安全要件だけでなく、非安全要件（表４のASIL決定表で“QM”と判定された要件）も考慮する。技術安全要件を実装するには、システム設計の検証可能性、機能安全を達成するための技術的能力、およびシステム統合中の試験実施可能性も考慮する必要がある。

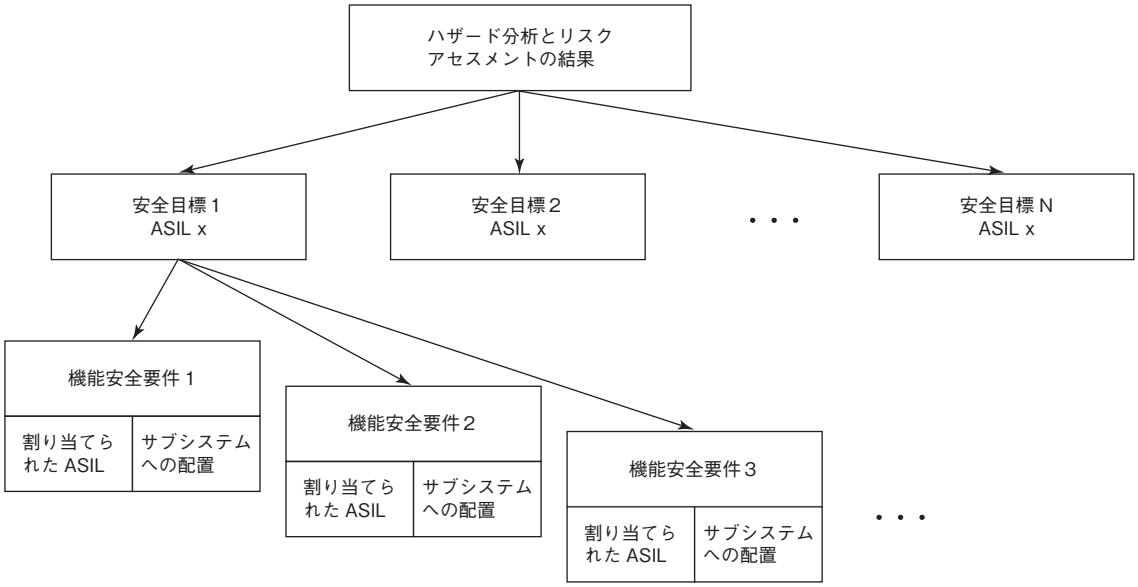


図３　安全目標と機能安全要件の階層構造

技術安全要件を実装に必要な仕様にまとめたものがシステム仕様である。

システム仕様の各要件はハードウェアとソフトウェアに分類し、更に詳細化する。また、ハードウェア・ソフトウェアインターフェース（HSI）を設計し、技術安全コンセプトと一貫性のとれたハードウェアとソフトウェアの分担と相互作用を明確化する。

また、システム仕様については、技術安全コンセプトに対する遵守と完全性を検証しなければならない。

7.5　ハードウェア安全要件の設定と機能安全のハードウェア開発

ハードウェアの安全要件は、ハードウェアに割り当てられた技術安全要件から得る。主に、ハードウェアの内部故障を制御する安全メカニズム(注8)の安全要件を保証する。

ISO26262が求めるハードウェア設計のポイントは、ハードウェアアーキテクチャ指標の評価と偶発的ハードウェア故障による安全目標逸脱の評価を定量的に実施することである。

ハードウェアアーキテクチャ指標とは、ハードウェアの偶発的故障に対するハードウェアアーキテクチャの効果（ロバスト性）を評価するための指標であり、規格が定める計算式で決まる定量的数値である。

偶発的ハードウェア故障による安全目標逸脱の評価とは、確率論に基づく安全性の論理的な裏づけをもって安全メカニズムの有効性を評価することである。

7.6　ソフトウェア安全要件の仕様化

ソフトウェア安全要件は、故障によって技術安全要件の逸脱を引き起こす機能を対象とし、ソフトウェアの実装・検証が可能なレベルまで詳細化して定義する。

ソフトウェア安全要件の仕様化においては、指定されたシステムとハードウェア構成、ハードウェア・ソフトウェアインターフェース、ハードウェア安全要件、時間的制約、アイテムから見た外部とのインターフェース(注9)、ソフトウェアに影響する車両、システムまたはハードウェアの各動作モードを考慮する。

また、ソフトウェア安全要件の仕様は、技術安全コンセプト、システム仕様、ハードウェア・ソフトウェアインターフェース仕様との適合性と一貫性を検証しなければならない。

（注８）　電気・電子ハードウェア、ソフトウェアやその他の技術によって、安全状態を確保するために障害を検出したり、故障を制御したり、または異常警告を発してリスクを回避したりする技術的解決策

（注９）　例えばペダル操作のブレーキシステムというアイテムと、その外部に位置づけられたABS（アンチロック・ブレーキシステム）とのインターフェースなど。

8.　機能安全のソフトウェア開発

機能安全への対応によって、ソフトウェアの開発は従来の方法と比べて何が変わるのか。本章ではこの点について述べる。

8.1　ソフトウェア安全要件のトレーサビリティ

ISO26262では、各ハザードのリスク低減を安全目標として設定し、それらが確実に実装／試験されたかどう

かを開発工程全体にわたって追跡できるようにすることを求めている。これを安全要件のトレーサビリティと呼ぶ。トレーサビリティは要件にID番号を付けて識別する。

ソフトウェア開発においては、ID番号を使って、ソフトウェア安全要件仕様（ソフトウェア機能仕様に相当）の安全要件、ソフトウェアアーキテクチャ設計仕様のソフトウェアコンポーネント、ソフトウェア単体設計仕様の関数を紐付けする（図４）。これによって安全要件の関連が明示的となり、上位要件に対する下位要件の抜け洩れや、関数の不具合が及ぼす要件への影響などを、検証や試験で効果的に発見して対処することができる。

要件のトレーサビリティ管理には、ソフトウェア開発などで用いられる要件管理ツール(注10)を使用する。

（注10） 米IBM社の「Rational DOORS」や米PTC社の「MKS Integrity」などがある。

8.2 設計手法

ISO26262では、より高いASILが求められるソフトウェアに対しては、より厳しい制約の設計手法が求められる。

ソフトウェアアーキテクチャ設計では、ASIL Dの場合、ソフトウェアコンポーネントの階層化、ソフトウェアコンポーネントサイズの制限、適切なスケジューリング、ソフトウェアコンポーネントの結束性、結合制限、割込み制限が必須要件である。

こうしたソフトウェア設計における各種の手法は、構造化設計手法などに出ている考え方で、特に目新しいものではないが、ISO26262ではASILレベルに対応して、必須となる手法の適用を明示している。

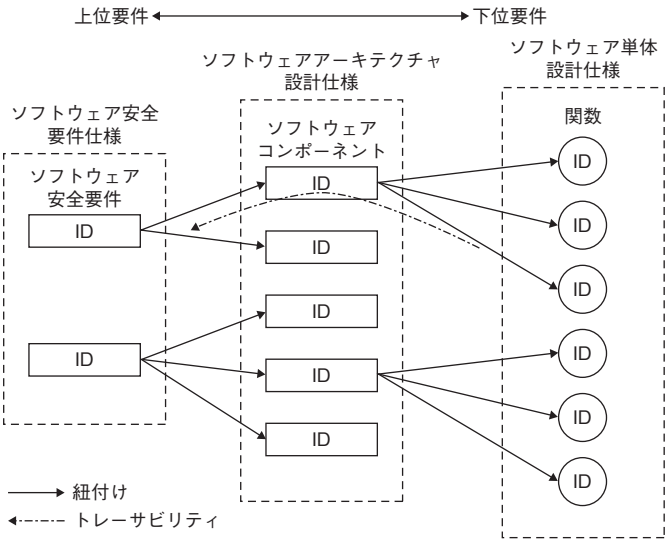


図４　ソフトウェア安全要件の継承とトレーサビリティ

ソフトウェアアーキテクチャレベルのエラー検出では、ASIL Dにおいて、入出力データの範囲チェック、データ妥当性のチェック、外部モニタリング、制御フロー監視、ソフトウェア冗長設計が必須となる。

ソフトウェアアーキテクチャレベルのエラー処理では、ASIL Dにおいて、エラー発生時の縮退機能、並列の冗長性が必須となる。

ソフトウェア単体設計では、ASIL Dにおいて、関数における１つの入口１つの出口、ヒープ領域など動的に確保されるオブジェクトの制限、変数初期化の実施、同一変数名の禁止、グローバル変数の使用制限、ポインタの使用制限、暗黙の型変換の禁止、隠れデータフロー／制御フローの禁止(注11)、無条件ジャンプの禁止、再帰呼び出しの禁止がある。

（各設計手法の詳細については、ソフトウェア設計技法等の書籍を参照されたい。）

モデリング設計について、ISO26262では深く触れていない。自動車のモデリング設計（モデルベース開発）では、MATLAB／Simlinkが代表的な手法とされるが、モデリング設計を行うことが、イコール、機能安全に寄与するわけではない。

モデリングに拠らないソフトウェア開発におけるコードの品質と同様、モデリング設計においてもモデルの品質を高めることが、それによって作られる製品（アイテム）の安全性に寄与する。

ISO26262では、モデルを設計する際のガイドラインについて、コーディングのガイドラインと組み合わせるかたちで、要件を定めている。

（注11） 隠れデータフロー／制御フローの禁止とは、制御フローが複数の意味を持つことを禁止することである。設計者の認識していない制御が実装されることにより、問題が発生するリスクが増加する。例えば、次の例が隠された制御フローに該当する。

【例】 設計者Aが関数Aの制御フローでグローバル変数Xを１ずつ加算していき、Xが10になったとき変数Yに１を代入するという設計を行ったとする。一方、別の設計者Bが関数Bで、本来なら依存関係がない関数Aの変数Xを、変数Zに１を代入するための判定条件として参照する制御フローを組むと、変数Xは設計者Aが認識していない隠れたフローに対しても影響を及ぼす。
即ち、この状況で設計者Aが変数Xに対して制御を変更すると、設計者に認識されることなく関数Bに対して影響が及んでしまう。こうした影響を回避するため、隠れデータフロー／制御フローは禁止されている。

8.3 検　証

検証（verification）とは「要求に対して“正しく”成果物が作成されたこと」の確認であり、レビューなどによって上位要件が抜け洩れなく下位要件に展開されているかという観点で実施する。

ISO26262では、ソフトウェアアーキテクチャ設計、ソフトウェア単体設計と実装（コーディング）において、ASILレベルに応じた検証が規定されている。

ソフトウェアアーキテクチャ設計の検証では、ASIL Aの場合、ウォークスルーだけでよいが、ASIL Dでは、インスペクション、動的箇所（実行可能モデル）に基づくシミュレーション、プロトタイピング、制御フロー／データフロー解析(注12)を必須手法としている。

ソフトウェア単体設計と実装（コーディング）の検証では、ASIL Aの場合、ウォークスルーだけでよいが、ASIL Dでは、インスペクション、準形式検証（モデルに基づくシミュレーションなどを用いた検証）、制御フロー／データフロー解析(注13)、静的コード解析が必須手法となっている。

(注12)　ソフトウェアアーキテクチャ設計の検証に使用する制御フロー／データフロー解析は、例えば、エラー検出時にシステムがフェールセーフ状態へ遷移できること、機能障害の場合、冗長機構が有効になることなどを確認する。この解析により、安全メカニズムの制御フローが正しい順序で行われていることを検証することができる。

(注13)　ソフトウェア単体設計と実装の検証に使用する制御フロー／データフロー解析とは、制御の流れやデータの流れを確認することである。ソフトウェア単体設計に基づいた制御フローやデータフローの実装が行われていることの解析を行う。

8.4 妥当性確認

妥当性確認（validation）とは「要求に対して“正しい”成果物が作成されたこと」の確認を行うことであり、試験によって要件を満足した成果物が作られているかという観点で実施する。

ASIL Dでは、ソフトウェア単体試験、ソフトウェア統合試験とも、要件に基づく試験、インターフェース試験、不具合混入試験、リソース試験(注14)、バックトゥバック試験(注15)が必須となる。

試験ケースの導出方法では、要件分析、同値クラスの作成と分析、境界値分析が、ASIL Dの必須要件である。

ソフトウェア単体試験の網羅率（カバレッジ）については、ASIL AがC0カバレッジ（命令網羅率）の100％実施でよいのに対し、ASIL DではC1カバレッジ（分岐網羅率）およびMC/DC（Modified Condition/Decision Coverrage)(注16)の100％実施が求められている。

ソフトウェア統合試験では、機能カバレッジ(注17)およびコールカバレッジ(注18)の100％実施がASIL Dで求められている。

(注14)　コードが実行可能か、ROM／RAM／レジスタなどのリソースがどの程度使用されているかを確認する試験である。

(注15)　モデルベース開発によって作成された実行可能なモデルのシミュレーションと自動生成されたコードに同じ試験ケースを与え、その実行結果を比較し、確認を行う試験である。

(注16)　すべてのコード中の条件や判定に対して、一度はコードを実行する。この際、判定の条件は判定の出力が独立するように試験を行う。条件の数をnとした時、全条件の組合せを試験するC2カバレッジ（複合条件網羅率）の試験ケース数が２のｎ乗個となるのに対し、MC/DCのケース数はｎ＋１個で済む。即ち、条件が多くなるほどC2カバレッジよりコスト的に有利となる。

(注17)　ソフトウェア全体の機能数と実行された機能の割合である。これにより、実行機能の網羅性を確認することができる。

(注18)　全体の関数（または機能）と実行された関数（または機能）の割合である。これにより、呼び出された関数の網羅性を確認することができる。

8.5 ソフトウェアツールの認定

ISO26262は、開発の使用目的として導入する各種のソフトウェアツール（以下、単にツールと表記）に対して信頼性評価指標を定め、その指標に適合すると認められるものが開発に使用できるとしている。

先ず、ツールが認定の対象になるかどうかの選別を行う。

それには、TI（Tool Impact)、TD（Tool error Detection)、TCL（Tool Confidence Level）という指標を使用する。

TIとは、特定のツールが誤動作した場合に、開発中の安全関連のアイテムまたはエレメントにおいてエラーが取り込まれたりエラーの検出に失敗したりする可能性である。TI1とTI2の２段階で評価する。TI1はツール誤動作によるアイテムまたはエレメントへの影響が全くないと論証できる場合に選択する。TI2はそれ以外の場合（即ち何らかの影響がある場合）に選択する。

TDとは、ツールが誤動作して、対応する誤出力を生じさせることを防止する手段、または、ツールが誤動作して、対応する誤出力を生じさせたことを検出する手段における信頼性である。TD1、TD2、TD3の３段階で評価する。TD1は、誤動作とそれに対応する誤出力を防止または検出できる信頼性の度合いが高い場合に選択する。TD2は、同様の信頼性の度合いが中程度の場合に選択する。TD3は、それ以外の全ケースで選択する。

こうして求めたTIとTDに対して、表５に示すマトリックスでTCLを決定する。

TCL1と判定されたツールは認定しなくても使用できる。

TCL2およびTCL3と判定されたツールは、次の４種類の認定方法を適用する。

表５　TCLの決定表

	TD1	TD2	TD3
TI1	TCL1	TCL1	TCL1
TI2	TCL1	TCL2	TCL3

①使用実績による信頼性の向上

ツールにおいて規格が定める使用実績に関する証拠が提供される場合に限り、使用実績による信頼度の向上が論証されるものとするなどの方法。

②ツール開発プロセスの評価

ツールの開発に適用される開発プロセスが、適切な規格に適合しているかを評価するなどの方法。

③ツールの妥当性確認

規格が定める判定指標を満たす妥当性確認を行う方法。

④安全規格に従った開発

ツールの開発が、例えばISO26262、IEC61508、RTCA DO-178（航空システムや装置の安全規格）といった規格に対応しているかを検証する方法。

これらの認定方法の適用は、ツールで開発するアイテムまたはエレメントのASILによって異なる。

TCL2の場合、ASIL A/B/Cでは①と②が、ASIL Dでは③と④が必須である。

TCL3の場合、ASIL A/Bでは①と②が、ASIL C/Dでは③と④が必須である。

なお、ツールをベンダーから購入する場合、ツールの利用者が認定を行うのは現実的に困難である。そこで、ISO26262に適合した第三者認証済みのツールを導入することが現実的であると言える。

8.6　保護機能による不具合の波及の防止

ISO26262では、特定のソフトウェアコンポーネントに不具合があった際、それが別のソフトウェアコンポーネントに悪影響を及ぼすのを防ぐための仕組みを「ソフトウェアパーティショニング」として規定する。この技術はASILの異なる複数のソフトウェアコンポーネントを、同一のMPU上で共存させるような場合に利用できる。例えば、ASIL AとASIL BとASIL Cのソフトウェアがあったとする。これをパーティションなしで１個のMPUと１個のリソース（共有メモリ）で動作させる場合、３つのソフトウェアは共に、最高ASILであるASIL Cを適用しなければならず、結果として必要以上に高い開発コストをかけなければならない。一方、リソースにパーティションがあれば、１つのMPUでも個々のソフトウェアコンポーネントに最適なASILを適用することができ、無駄な開発コストを避けることができる。

ソフトウェアパーティショニング技術にはいろいろあるが、自動車用MPUにソフトウェアパーティショニング技術を適用する場合、現時点ではコスト面から、MPUのメモリ保護ユニットを利用したOSによる方法が最も有望であるとされている。

8.7　プロセス改善

組み込みソフトウェアの分野では、CMMI、Automotive SPICEといった各種プロセス改善規格への取り組みを継続的に実施することが求められる。

ISO26262においてもプロセス改善が求められているが、これは、基本的に既存のプロセス改善の延長線上にある。CMMIのレベル２、Automotive SPICEのレベル３程度の認証がとれていればよいとされる。

ISO26262ではPart ２機能安全の管理（図１）の中で、技術的観点から成果物を検証する確証レビュー、プロセス的観点から機能安全活動の実装状況を確認する機能安全監査、アイテムの安全性について機能安全の適合性を評価する機能安全アセスメントの３点セットをまとめた確証方策に対して、ASILに対応した独立性をもって実施するよう規定している。

独立性は４段階が定義されており、最も軽いレベルは「確証方策を異なった人物で実施したほうがよい」という程度であるが、最も重いレベルは「異なった部署あるいは組織の人物（即ち、第三者）によって実行されなければならない」となる。確証方策の内容にもよるが、ASILレベルが高ければ、独立性も高いレベルが求められる。

なお、第三者の組織については、品質保証部といった社内の独立した部門でもよいし、社外の第三者認証機関でもよい。営業戦略上、名の通った第三者認証機関に監査を依頼するサプライヤやツールベンダーが多い。代表的な第三者認証機関の１つに、ドイツ・TÜV SÜD（テュフ・ズード）社がある。

9. 安全文化

ISO26262では、組織として安全活動を保証する仕組みの構築を求めており、そのためには、組織として安全な製品開発が継続できるプロセスインフラの構築が求められる。プロセスインフラの構築にあたっては、「人は誰でも間違いをおかす可能性がある」という性悪説に立ち、仕組みの構築を考える必要がある。

そこで登場する重要な概念が「安全文化（Safety Culture）」であり、ISO26262は、組織（および個人）が安全を重視する企業の風土を構築していくことである。

安全文化とは、組織として安全に関してあるべき姿（目的に対して十分な効果があり、関係者が内容を十分に理解して合意が得られている状態）の規則やプロセス

が構築され、適切な教育や啓発活動によって個々の担当者の中に「安全に関する共通の常識」として浸透させるべきものであり、これは一朝一夕にして定着できるものではない。そのため、ISO26262は、組織の標準プロセスを定義し、各プロジェクトの特性に合わせたテーラリング（標準プロセスの仕立て直し）を行う仕組みと、継続的なプロセス改善活動を求めている。

ISO26262はプロセスの構築が最終的な目標ではない。プロセスが組織の求める目標を達成することによってプロセスの構築が意味を持つ。

10. ISO26262推進に必要な組織体制

ISO26262推進に必要とされる組織体制例を図５に示す。

これは、CMMIやAutomotive SPICEの体制にセーフティマネジメントチームを加えた体制となっている。

この体制は、規格の中で直接要求されていることではないが、ISO26262対応を推進していくためには、推進体制の構築が不可欠である。

CMMIやAutomotive SPICEによるプロセス改善では、ソフトウェア改善推進組織となるSEPG（Software Engineering Process Group）を置く。ソフトウェア開発のプロセスに限定することなく、プロセス改善を推進する組織をEPG（Engineering Process Group）とも呼ぶ。このEPGがISO26262の推進役として果たす役割は、機能安全に関する活動を含めた開発活動全体に対し、適切な組織標準プロセスを構築し、それを実施する各担当者に必要な教育をおこなうことである。また、さらに組織標準プロセスを維持するために、必要なデータや情報を収集してプロセスを改善することである。

11. 分散開発（開発委託）の要件

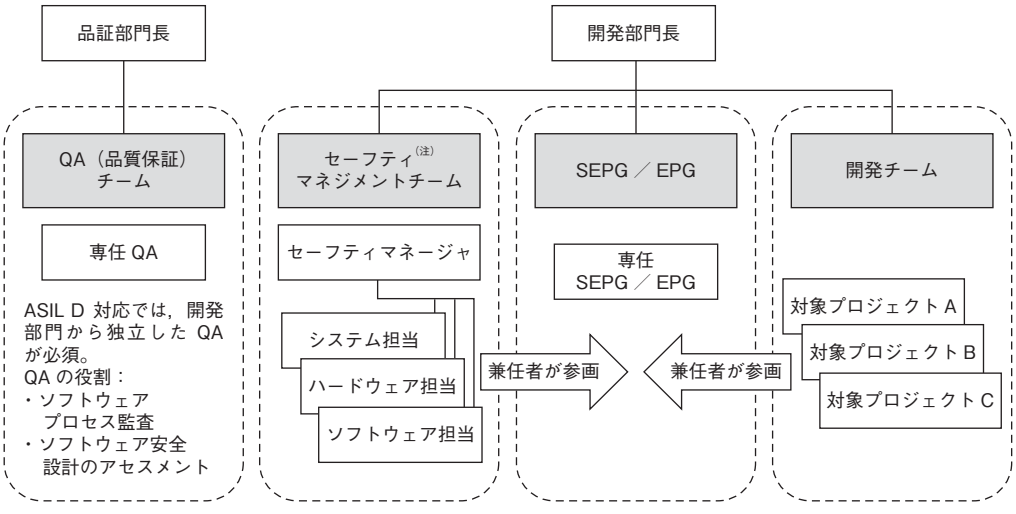
機能安全を適用するアイテムやエレメント（ソフトウェアを含む）を開発委託する場合、受託するサプライヤ（ソフトウェア受託会社を含む）もISO26262が求める要件を満たさなければならない。

開発を委託する前には、開発委託者とサプライヤとの間で開発インターフェース合意書（DIA（Development Interface Agreement））を取り交わす必要がある。DIAには、開発委託者とサプライヤの双方のセーフティマネージャ、安全ライフサイクルのテーラリング(注19)、開発委託者とサプライヤそれぞれの活動とプロセス、双方で交換する情報や活動などが含まれる。

開発活動中のサプライヤは、進捗、リスクを増大する課題、不具合を開発委託者へ報告しなければならない。サプライヤによって満たすことのできない安全要件がある場合は、安全コンセプトを再検討し、安全要件を見直す必要がある。

アイテムの安全性に影響を与える可能性のある変更は、影響分析を行うために開発委託者とサプライヤが互いに通知しあう必要性がある。そして、安全の妥当性確認を開発受託者が行うのか、サプライヤが行うのかを決めておかなければならない。

（注19） ISO26262が示す安全ライフサイクルとは、製品のコンセプトから開発、生産、運用、廃棄に至る製品ライフサイクル全体を通じた安全管理のライフサイクルである。
安全ライフサイクルのテーラリングとは、ISO26262で定められた安全ライフサイクルを組織の特性に合わせて仕立てなおす行為である。組織によっては各フェーズの活動を統合したり、分割したり、反復したり、または別のフェーズで実施したりする。



（注）小さな部門やプロジェクトでは、セーフティマネジメントチームを開発チームの中に含めてもよい。

図５　ISO26262推進に必要な組織体制（例）(3)

その他、ISO26262では分散開発に関するいくつかの要件を定めている。

12. むすび

ISO26262は2011年11月に制定されたが、特に欧州や米国などの海外自動車メーカ各社は、早くから機能安全に取り組んできたこともあって、ISO26262の要件に対応した成果物やプロセスをサプライヤへ要求している。また、国内の自動車メーカ各社もISO26262に基づく安全要件を求めつつある。そうした状況の中で自動車メーカに部品を納入しているサプライヤの中には、ISO26262の要件を満たすため、いち早く体制を整え、規則を整備し、規格に基づくプロセスを構築して第三者機関による認証を取得したところもある。

当社も、安全にかかわる車載機器の制御ソフトウェア開発の請負業務を担当する部門において、今後、ISO26262対応の整備が必要になるものと思われる。

参考文献

⑴ 「クルマの電子安全始まる　ISO26262を超えて」日経エレクトロニクス2011.1.10号

⑵ 「ISO26262におけるハザード分析およびリスクアセスメント」自動車技術，Vol.66，No.9（2012）

⑶ 「ISO26262実践ガイドブック［入門編］」日経BP社，第１版１刷