ソフトウェア開発におけるヒューマンエラーの排除

（中武、山下）

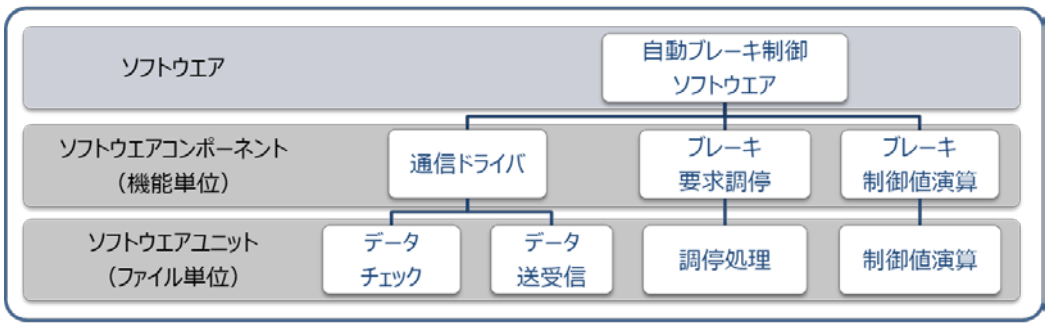
前回までのメルマガでは、昨年、今年に出版された「ISO 26262 2nd 実践ガイドブック 入門編～機能安全の正しいアプローチ～」「ISO 26262 2nd 実践ガイドブック システムエンジニアリング編」を紹介いたしました。

2021 年 3 月 31 日に出版予定の「ISO 26262 2nd 実践ガイドブック ソフトウェアエンジニアリング編」では、前述のガイドブックに引き続き、プロセスアプローチと安全設計の視点で、安全ライフサイクルにおける各ソフトウェアエンジニアリングの活動の流れと活動内容を具体的な事例（自動ブレーキ制御機能）を用いて解説しています。解説範囲は、ソフトウェア開発における ISO 26262 の Part-6、設計フェーズにおいて関連する安全要件管理や安全分析などの Part9 です。特にソフトウェアの開発における人の作業は設計ミスや実装ミスなどのヒューマンエラーが混入しやすいため、そのヒューマンエラーに起因して発生するシステマティックフォールトを考慮することが必要となります。

本メルマガでは、ソフトウェアアーキテクチャ設計における解説の一部を紹介いたします。

ソフトウェアアーキテクチャ設計では複雑な設計により発生するヒューマンエラーに起因するシステマティックフォールトのリスクが多く潜在しています。そのリスクを軽減および回避する手段として、実施する活動と活動の流れを予め手順書として整備することや、従来のソフトウェア開発における基本的なアプローチである構造化設計の考えに基づく設計原則を適用するなどのプロセスアプローチがあります。今回はその設計原則の中から「ソフトウェアコンポーネントの階層構造」を例に説明します。

設計原則「ソフトウェアコンポーネントの階層構造」では、下図のようにソフトウェアを一定の機能単位となるソフトウェアコンポーネントやソフトウェアの実装が可能な粒度でファイル単位となるソフトウェアユニットで分割し階層構造を定義することで、ソフトウェア全体を容易に理解することができます。ソフトウェアコンポーネントやソフトユニットという分割の粒度は ISO 26262 で定義されているわけではなく、開発現場で定義する必要があります。ソフトウェアユニットについてはソフトウェアユニット検証で静的検証および動的検証がソフトウェアユニット単体で実施できる粒度が適切であると言えます。階層数に関してはソフトウェアが小規模な場合はソフトウェアからソフトウェアユニットまで分割することもあり、ソフトウェアの規模に応じて決定する必要があります。



ガイドブックでは上記で紹介したヒューマンエラーを防止するためのプロセスアプローチと FTA/FMEA などの手法を用いた安全分析結果から危険事象を検出し安全機構を施す安全設計アプローチについて解説しています。

開発現場の皆様が、ガイドブックで解説しているソフトウェアエンジニアリング領域のプロセスアプローチと安全設計を理解し、機能安全製品の開発で実践できるようになるためのトレーニングを準備しています。弊社ホームページ、または次号以降のメルマガにて案内いたします。機能安全対応でお困りの際は、弊社コンサルティング事業部にお気軽に、ご相談下さい。

2021/3/26 中武 俊典、山下 祐太朗