機能安全時代のソフトウェア組み込み型製品開発のプロセス定義

2011/10/27

パナソニック エレクトロニックデバイス(株)
技術統括部　機能安全推進チーム
安倍秀二

Panasonic
ideas for life

本日の発表の内容

- 初めに
- 背景
- ISO26262のプロセス要求
- 安全ライフサイクル
- 課題
- システムで重要なこと
- 機能安全対応プロセスの構成
- 機能安全プロセス体系
- フェーズとDCP
- 基幹システムとの整合
- 役割定義の追加とスキル
- 各フェーズでの安全分析
- 安全の作り込みと確認
- 今後の課題

Panasonic Electric Device

初めに

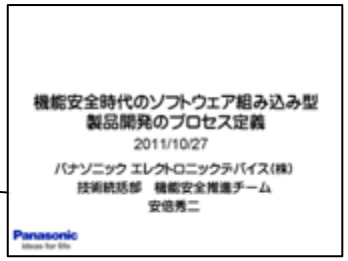
PEDの商品

PEDのソフトウェア

- PED全体から見るとデバイス部品の比重が多く、ソフト搭載部品は少ない。
- しかし特に車載で、顧客からの品質要求（開発プロセスを含む）が非常に高い
- プロセスアプローチで開発に柔軟に対応
- 強力なトップダウンでグローバルにプロセス改善やプロセスの標準化を推進

SJ2010

SJ2011



システム製品監査

セッション2C
品質保証

私のこと

- ～2002年…エレキ、ソフトの開発を担当
- 2002年…プロセス改善を担当する
- 2007年…現在の部署に移動
- 2008年…CMMIレベル3達成
- 2010年…これまでの成果を発表
- 2011年…製品全体のプロセス開発。
  機能安全の推進担当

背　景

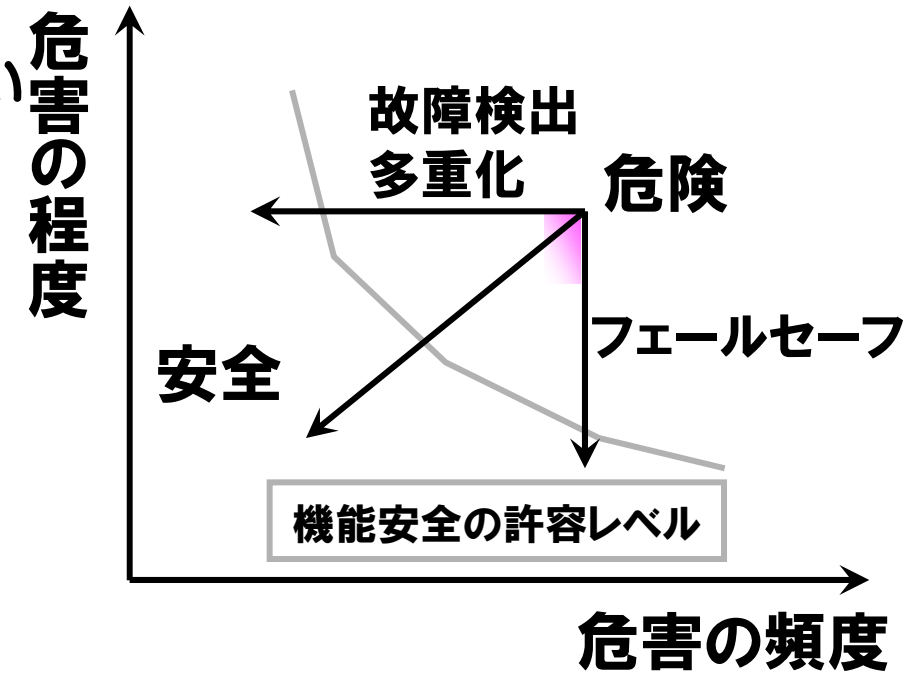
- 車の機能の複雑化によるソフトウェア、電子回路の比重の高まり
- 安全要求の高まり
- 機能安全規格の規格化とOEMからの遵守要求

挿入図：テュフズードジャパン　竹市氏「機能安全認証の現状」から引用

機能安全とは

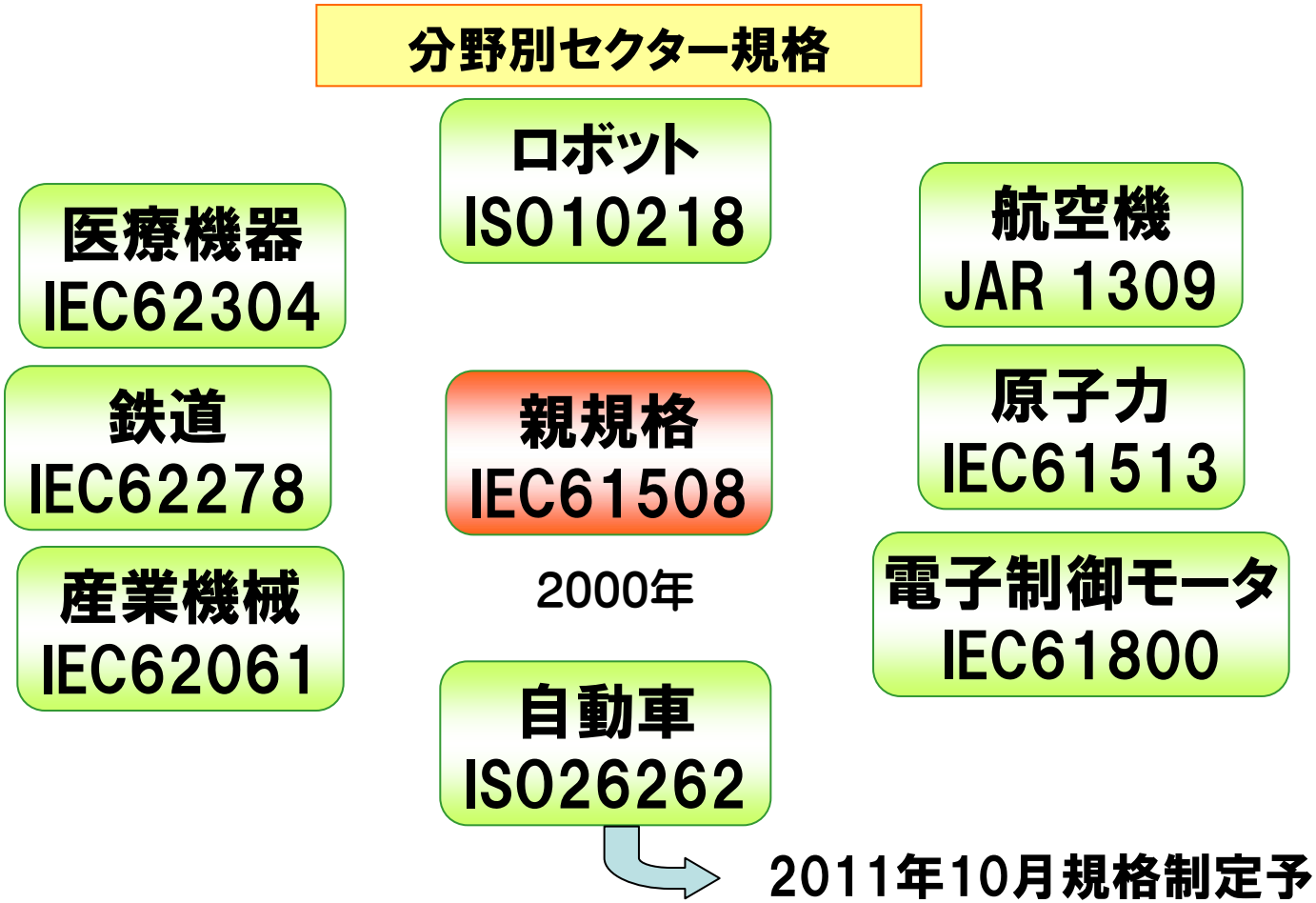
- 致命故障に対する一般的な設計の考え方（自動車分野）
  - 人命に危険を及ぼす故障や法規制に不適合となる故障モードは、一般的に車両寿命に至るまで故障してはならない
- 機能安全の考え方
  - 故障0（リスク0）はあり得ない

フェールセーフや多重化の採用などのいろいろな技法の適用により、リスク（危害の程度×危害の頻度）を許容レベルまで下げることで安全を確保する設計思想



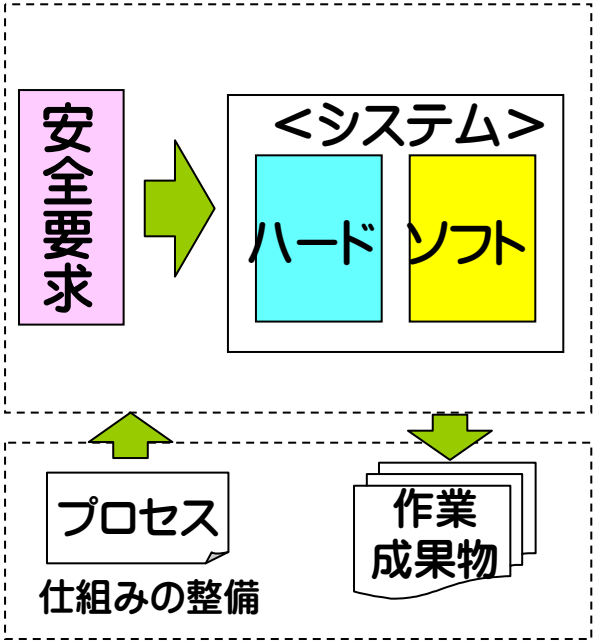
機能安全規格群

プラントを中心とした電気電子機器の機能安全規格IEC61508から
自動車用向けのISO26262が作成された。
（自動車はプラントよりもリスクシナリオが複雑（走行環境・人・安全防護策等））



機能安全規格～ISO26262とは

- 車載システムに搭載された電気／電子制御システム(ECU)の機能欠陥に対する安全確保の仕組み
- IEC61508を頂点とする機能安全規格体系の車載システム専用の規格
- 市場で起こる故障をシステマティック故障(設計に起因)とランダムハードウェア故障(ハードウェア部品故障に起因)に分けて、故障を回避する安全設計活動の導入とその説明責任
- ASIL(車載安全度レベル)の概念を導入し、車の実使用場面での危険度合いや発生度合いに基づいて活動内容のレベルを定義



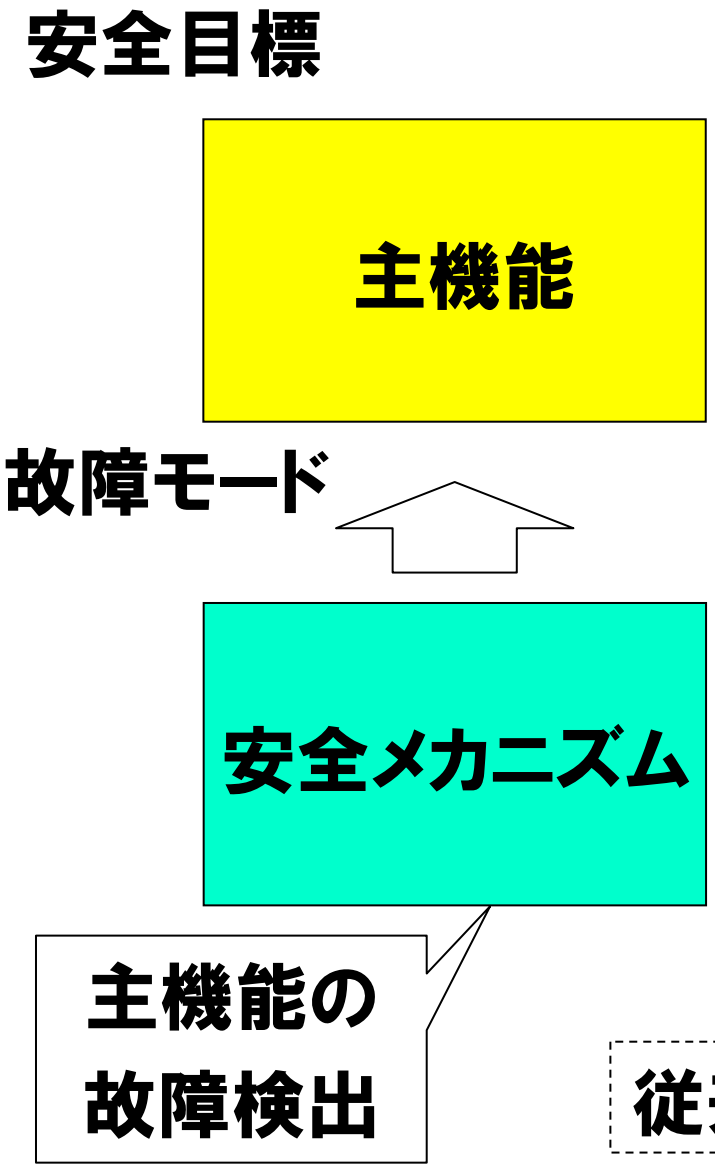
第三者による確認

●プロセスの遵守
●成果物レビュー
●成果物審査

ISO26262のプロセス要求

- 製品開発のプロセス定義
  - 製品開発全体の安全ライフサイクル定義を要求
  - 品質システムとしてISO9001/TS16949が必要
  - AutomotiveSPICEが前提（ENG、SUP）
- ASIL が付与されたら特別な要求
  - 詳細な内容
    - 手法、設計原則、評価指標　など
- 安全活動実施の証拠
  - 安全ケースによる論拠

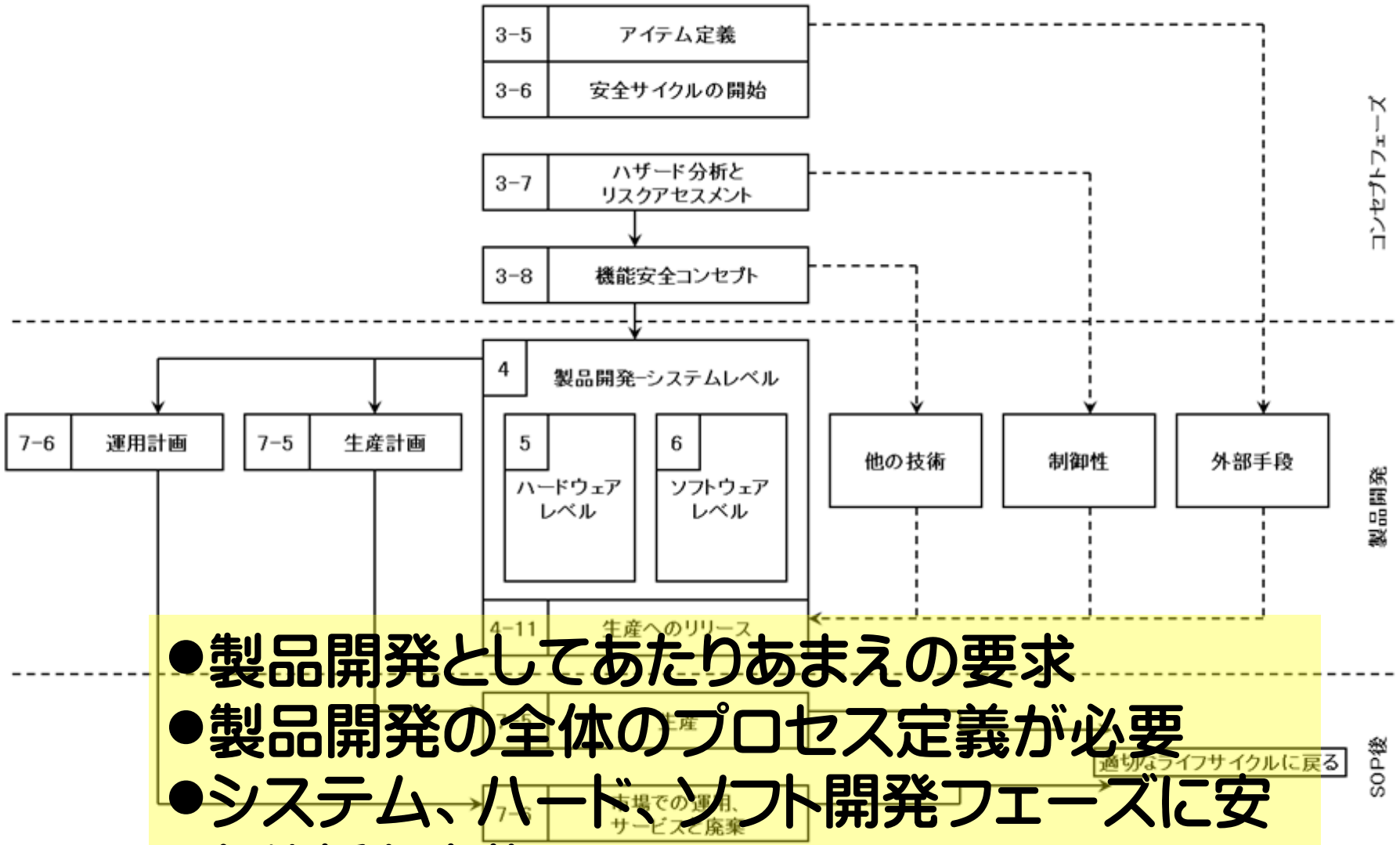
機能安全に対応する設計思想



- システマティック故障
  - 厳密なプロセス定義による設計エラー混入の防止
  - 設計原則などの遵守
- ランダムハードウェア故障
  - システム全体を主機能の故障検出のカバー率で評価
  - ハードウェア部品の故障率による故障リスクの評価

従来のフェールセーフの考え方を発展

安全ライフサイクル



- ●製品開発としてあたりあまえの要求
- ●製品開発の全体のプロセス定義が必要
- ●システム、ハード、ソフト開発フェーズに安全分析を実施

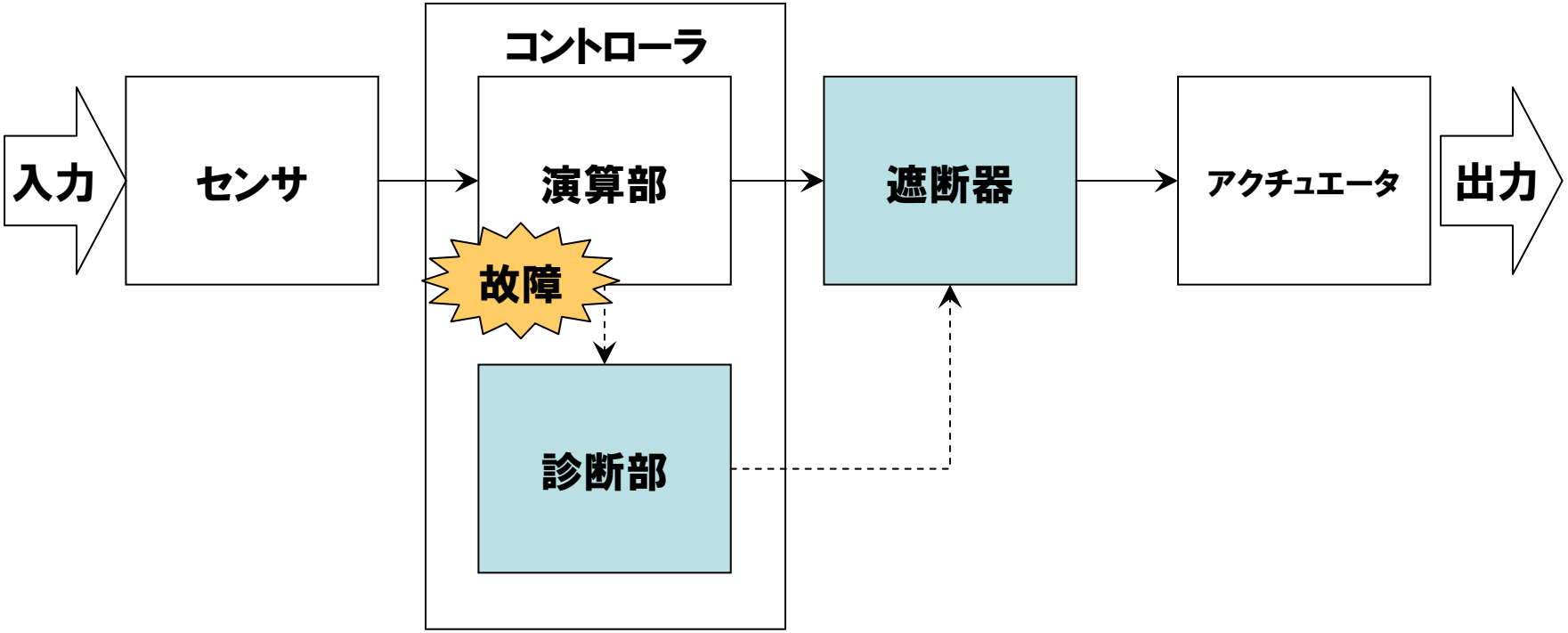
課　　題

- プロセスをベースに仕事をしていくというマインド
  - ソフトウェア
    - 2000年以降のプロセス改善により、プロセス志向が浸透してきた
    - 社内のソフトウェアプロセスガイドライン、ISO12207、ISO15504、CMMIなど環境が整っている。
  - ハードウェア（エレキ）
    - 社内の基幹システムに沿った製品開発フローはあるが、活動に関する詳細なプロセス定義が不十分であった。
    - ソフトは工学的な体系（SWEBOKなど）があるがエレキは不明
- メカ、エレキ、ソフトの順で作業の一般化が困難
  - 製品に依存

26262はシステム、エレキのプロセス定義も必須

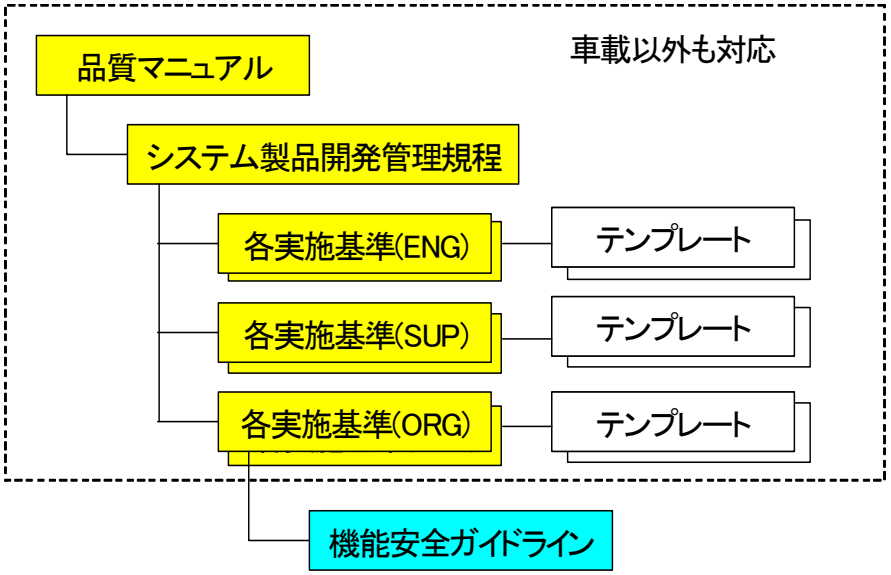
ソフトウェア組み込み型製品で重要なこと

- システム設計の善し悪しがコストと性能を決める
  - 機能安全要求の実装はバランスが大切
  - 上流作業はソフトとハードの共同作業
  - ソフト、ハードの知識が必要
- 下流のソフト・ハード結合後の妥当性確認

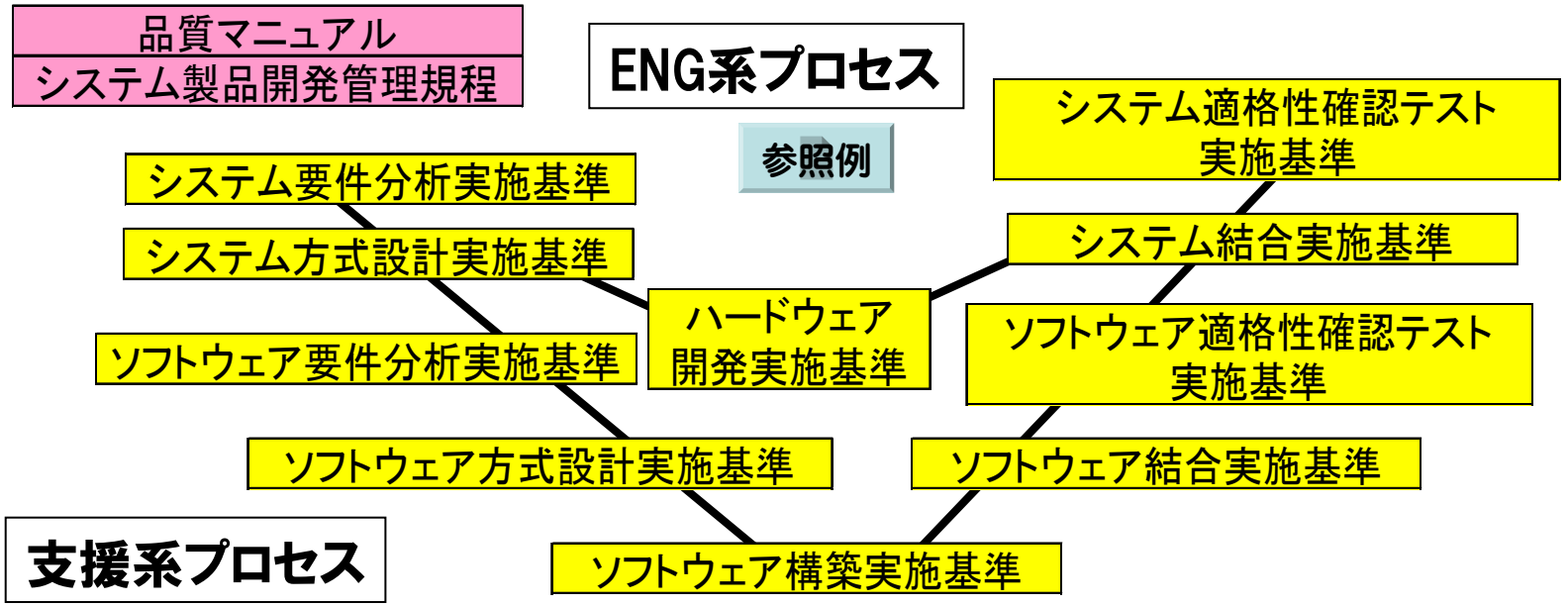


機能安全対応プロセスの構成

- 4階層の構成
  - 品質マニュアル
  - 全社基幹システムを基本とする製品開発サイクルの定義
  - システム、ハードウェア（エレキ）、ソフトウェアに関するアクティビティをプロセス定義
  - 規格要求と解説を記述した機能安全ガイドライン
- 規格要求である確認手段の実施（レビュー、監査、審査）の活動を製品開発サイクルのデシジョンポイントに関連させて定義
- お客様の開発に合わせ、柔軟に対応できる反復開発が前提
- 機能安全要求製品以外も対応可



機能安全プロセス体系



支援系プロセス
プロジェクト計画策定実施基準
プロジェクト管理実施基準
要件管理基準
構成管理実施基準
問題解決実施基準
変更管理実施基準
外注管理実施基準
デザインレビュー実施基準
品質監査実施基準
製品審査実施基準
組織管理基準

機能安全ガイドライン	ガイドライン
安全要件定義ガイドライン	安全分析ガイドライン
システム設計ガイドライン	機能安全計画策定ガイドライン
システムテストガイドライン	スキル認定ガイドライン
ソフトウェア設計ガイドライン	ソフトウェアツール 認定ガイドライン
ソフトウェアコーディングガイドライン	
ソフトウェアコード分析ガイドライン	ソフトウェアコンポーネント 認定ガイドライン
ソフトウェアテストガイドライン	
MBD開発ガイドライン	ハードウェアコンポーネント 認定ガイドライン
ハードウェア設計ガイドライン	

安全の作り込みと確認方策

- 確認方策
  - 確認レビュー
    - DRについては、技術レビューとマネジメントレビューを明確に分離
    - 独立の組織による第三者レビューを導入
  - 機能安全監査
    - システム製品開発プロセスの実施を確認
    - マネジメントレビューのインプットにする
  - 機能安全アセスメント（製品審査）
    - 安全要求が確実に実装されたかを確認

フェーズとDCP

- 全社の基幹プロセスのフェーズを安全ライフサイクルに割り当て
  - 企画フェーズ、構想フェーズ、計画フェーズ、開発フェーズ、生産準備フェーズ、生産発売フェーズ
- 確認手段として
  - 技術レビュー、第三者レビュー、システムプロセス監査、製品審査に割り当て
- DCP（デシジョンポイント）
  - プロジェクト開始（チャーター）、プロジェクト構想、計画策定（初期）、品質ゲート（AQ0,1,2）をDCPとする
  - 開発フェーズを構想設計サブフェーズ、システム設計サブフェーズ、設計・実装・テストサブフェーズに分割
    （※計画は反復計画であり、プロセス実施は柔軟。DCPはそれぞれ目的に応じた内容を確認）
  - マネジメントレビューとして、技術部門内品質確認ゲート（DR0, 1, 2）を割り当て。各プロセス成果物の技術レビュー、第三者レビュー、システムプロセス監査の結果を踏まえて移行承認。

社内基幹システム

各フェーズでの安全分析

- 構想設計
  - 機能安全コンセプト(MF/SM)
  - 初期のアーキテクチャ

  } 安全コンセプトの作り込み

- システム
  - 技術安全コンセプト(MF/SM)
  - 技術安全要件
  - 従属故障の評価

  } 安全コンセプトを技術仕様に落とし込み

- ハードウェア
  - 故障率とアーキテクチャの定量評価
  - ランダムハードウェア故障の回避
    - シングルポイントフォールト
    - 潜在フォールト

  } 全体システムの最適性を評価

- ソフトウェア
  - モジュールの独立性
  - 安全メカニズム

  } 自分の故障を他に影響させない

役割定義の追加とスキル

- システム開発領域の実施者としてSysENGを新設
- 理想はソフト、ハードの両方のスキルを保有し、開発の推進ができる人（目指したい）
- 現実はソフト技術者（SwENG、SwTENG）とハード技術者（HwENG、HwTENG）の共同作業として、ソフト、ハードの最適化、抜けのない開発を推進

まとめ

- 機能安全を推進するに当たって気がついたこと
  - 上流設計が重要
    - システム（製品）のコスト、性能はシステム設計及び、さらに上流のコンセプト設計が重要。
    - ソフトウェアは安全コンセプトを実現するに当たりKeyとなる。
      - メイン機能の故障検出はソフトウェアに期待。そのためにはソフトウェアのアーキテクチャ設計が重要
  - ソフトウェアの安全設計
    - 不具合の源は常に潜んでいることに気づくべき。リソースの問題（データ化け、期待しない割り込みなど）に対応した設計が重要。
  - SW/HWの協調設計
    - 特に組み込みは、SW/HWの両方の知識が必要。HWの特徴を引き出すのはSWの役目であることが多い。
    - マルチプレイヤーを目指そう
  - すべての設計には思想と根拠が必要
    - 個人にあるものは明示を
    - どうなっているか？なぜそうしたか？～説明する/できる責任

今後の課題

- 機能安全監査（システムプロセス監査）
  - 受ける側のプロセス教育。
- 機能安全アセスメント（製品審査）
  - 安全の視点を置いたしくみの実践
- 機能安全マネージャーの育成