車載向け機能安全規格ISO26262の車載以外へのソフト開発への応用

～もし家電メーカーのプロセス改善担当者が
車載向け機能安全規格「ISO26262」を読んだら～

2011/10/27
パナソニック アドバンストテクノロジー(株)
エンジニアリングセンター
川崎　雅弘

はじめに

■概要

機能安全対応ソフトウェア開発とはどのようなものか。

客観的に信頼性／安全性を満たせると言えるために、何をすれば良いか。

車載向け機能安全規格ISO26262対応プロセスの構築で得られた知見を、**家電中心の開発者の視点**で報告する。

【注意事項】

・本発表のベースとなるプロセスは、車載関連の開発の実態（カーメーカーの要求）も含みます。
このため、ISO26262規格で求められている要求と異なる（厳しい）部分もあります。
正しい規格上の要求については規格書をご参照ください。

本日の内容

■背景

- 従来開発の課題

■車載系と家電系のソフトウェア開発との違い

- 機能安全とは
- 機能安全のための活動
  - ー安全性
  - ー良い設計
  - ー確認

■まとめ

- ISO26262規格の車載系以外への応用の可能性

■背景

- 従来開発の課題

■車載系と家電系のソフトウェア開発との違い
- 機能安全とは
- 機能安全のための活動
  - ー安全性
  - ー良い設計
  - ー確認

■まとめ
- ISO26262規格の車載系以外への応用の可能性

従来の開発の課題

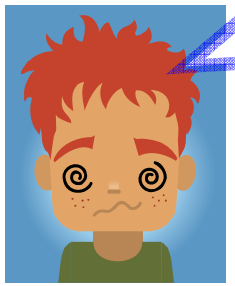
安全性/信頼性のための活動をしているのに、それでも市場クレームが発生

部品レベルの信頼性向上では対応困難な、エンドユーザー視点での問題が対象となってきている

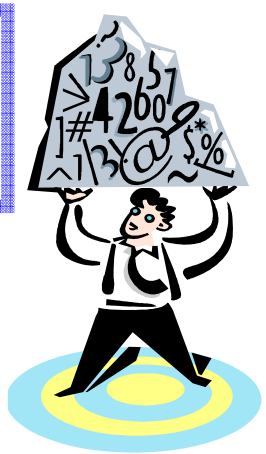
世界基準に基づいた説明可能な安全性/信頼性の必要性が高まる

現場の悩み

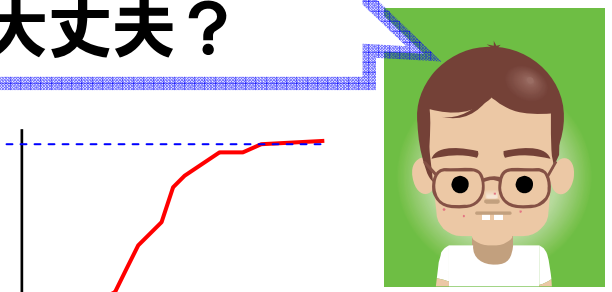
「市場不具合ゼロ」という目標を言われても．．．



レビューやテストを
どれだけすればいい？



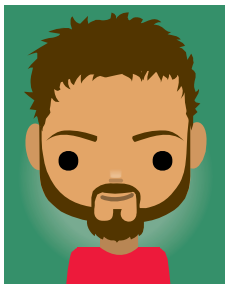
これで出荷しても
大丈夫？



バグ収束曲線

コストや期間は無限にかけられない

信頼性／安全性を証明できる「よりどころ」が欲しい



先人に学ぼう！
高い安全性が求められる
車載分野はどうしている？

ちょうど、機能安全規格
ISO26262が制定される！

Panasonic ideas for life

■背景
・従来開発の課題

■車載系と家電系のソフトウェア開発との違い
・機能安全とは
・機能安全のための活動
　－安全性
　－良い設計
　－確認

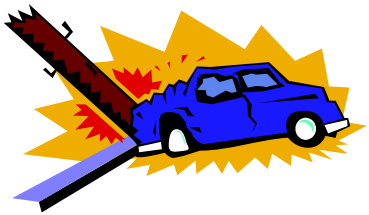
■まとめ
・ISO26262規格の車載系以外への応用の可能性

機能安全とは

機能安全とは、

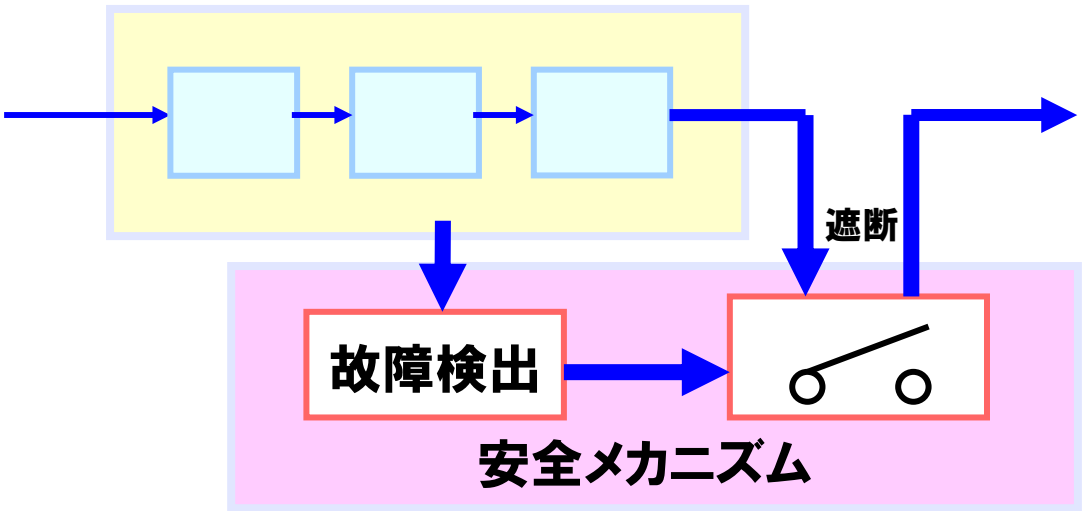
「機能により人の安全を確保する」

という考え方。



- 故障が発生しても**安全状態に移行させる機能**（安全メカニズム）を実装
  - 部品単位での安全対策にいくらコストをかけても、故障を完全になくすのは極めて困難
  - 安全メカニズムをシステムとして**確実に実装**することでユーザーレベルの安全性を担保

実装例



機能安全のための活動

故障しても危険状態に移行しないことを論理的に説明できること

⇒ 科学的手法/根拠による分析、明瞭な安全系アーキテクチャ設計で安全性を実現

⇒ それらを厳格なプロセス定義で確実に実施し、エビデンスやトレーサビリティで確認

安全性(＝危険状態に移行させない)

信頼性(＝欠陥がないことの担保)

ハザード分析＆リスクアセスメント	安全目標定義	システムの安全分析と安全要件定義	設計／実装	テスト
・運転状況 ・機能失陥、ハザード ・危険事象 ・回避行動 ・人的被害	・ASIL ・安全目標 ・安全状態	・安全メカニズム ・システマチック故障分析 (FTA/FMEA) ・ﾊｰﾄﾞｳｪｱﾗﾝﾀﾞﾑ故障分析 (故障率) ・安全要件の配置	・安全ﾒｶﾆｽﾞﾑ実装 ・設計原則 ・コーディング原則 ・ｶﾞｲﾄﾞﾗｲﾝ遵守確認	・テスト手法の規定 ・カバレッジ計測 ・車両環境テスト ・安全目標のテスト

コンセプトフェーズ → システム開発 → ハードウェア開発 / ソフトウェア開発 → システム開発

■考え方は車載系以外にも適用できる

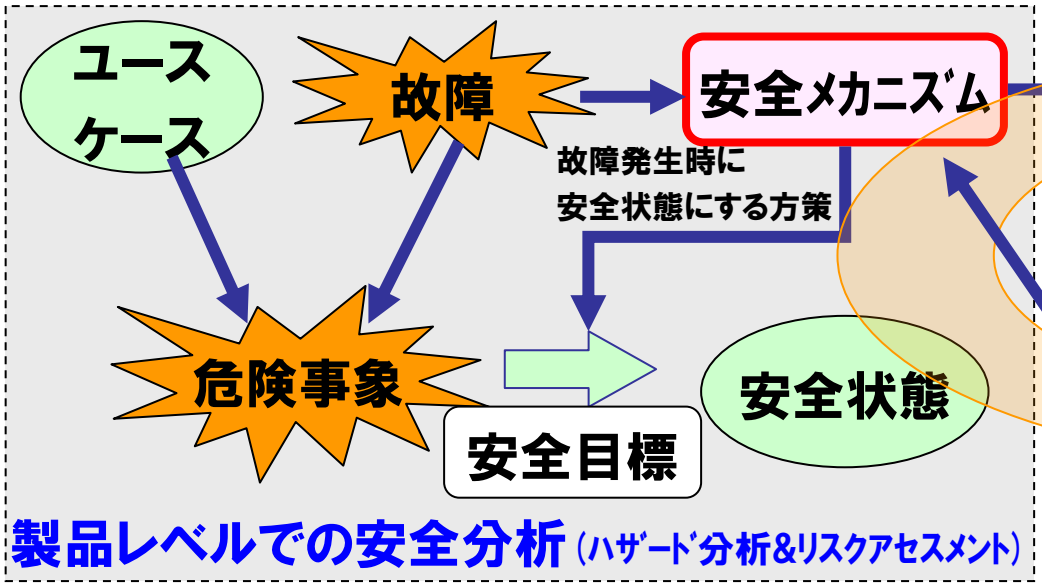
ソフトウェア観点でのポイント:①安全性、②良い設計、③確認

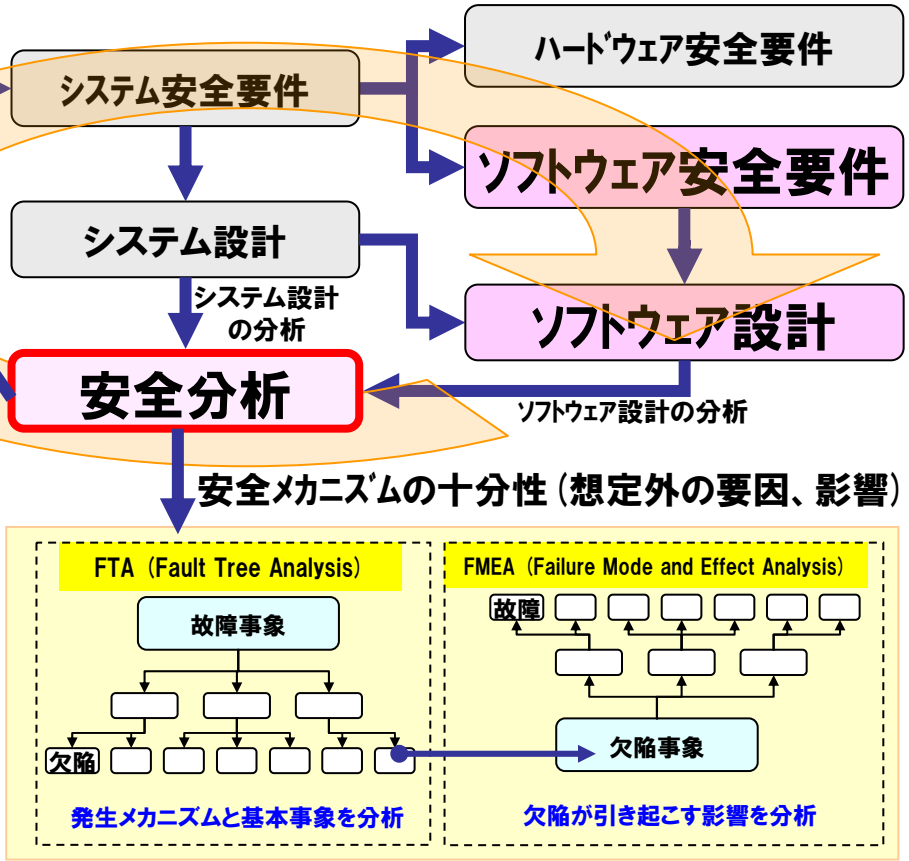
Panasonic ideas for life

①ソフト安全要件は製品レベルで考える

製品レベルでのユースケースと故障の仕方から対策を考え、ソフト要件とする

― お客様の被害を最小限に抑える(安全状態)ように「どのように壊れさせるか」を定義





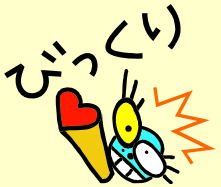
■ソフトウェアレベルでの安全活動

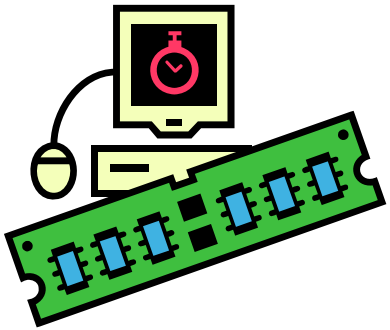
- **開発フェーズ全体を通じての安全分析**
  - ― 必要ならば安全メカニズムを再検討
- **安全要件の確実な実装**
  - ― 信頼性を担保することで製品の安全を実現

・科学的手法／根拠に基づく分析

事例:安全メカニズムの十分性

そこまで考えないといけないの！？





例:ソフトウェアでのROM/RAMチェック

[現状]電源ON時にチェック、異常を検出したら起動させない

⇒ ROM/RAMは**動作中にも故障**し得る。

定期的なチェックの必要はないか？必要ない理由は？

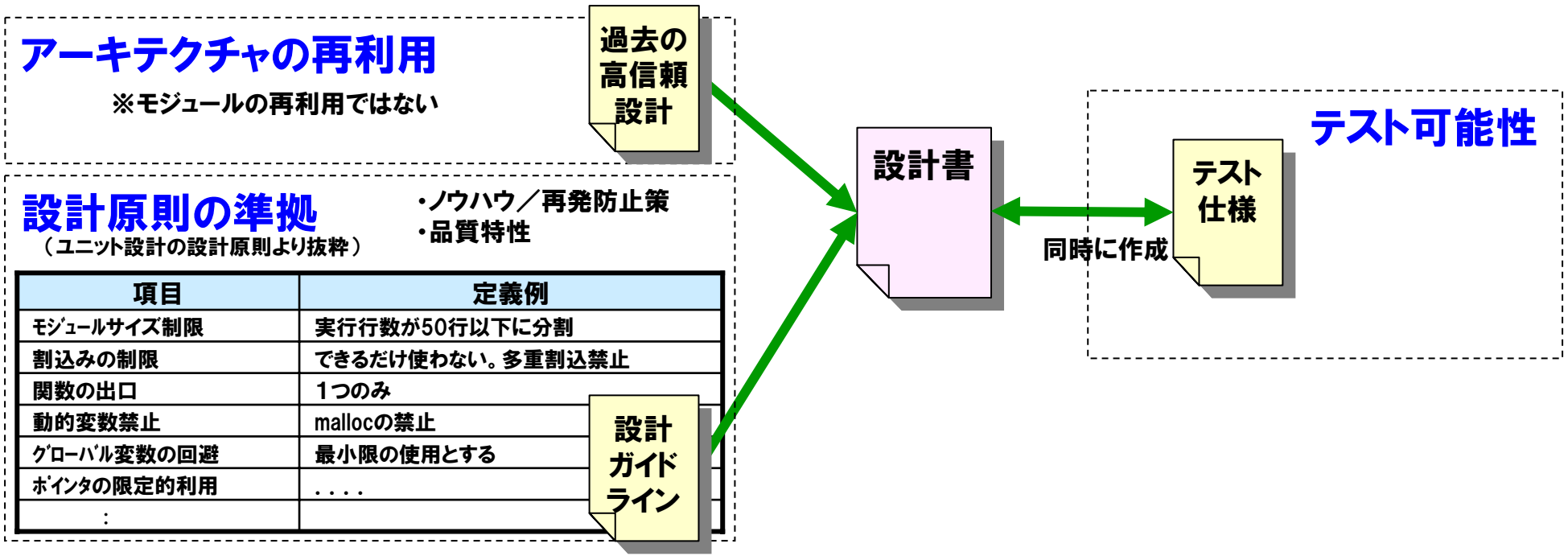
■故障発生時について、3つの「もし」を考えてみる

- もし、その安全メカニズムが無い場合、製品レベルでどのような影響があるか
- もし、故障が起こった場合、その安全メカニズムだけで影響を回避できるか
- もし、想定した状況下以外で、その故障が起こった場合も回避できるか

- 開発者の経験に基づく安全メカニズムではなく、
  これらを**説明できるように**安全分析をプロセスに組み込む

②欠陥の入り込み難い設計の実現

過去の「良かった設計」から、「良い設計」を実現する
- 欠陥の入り込む余地を最小とし、かつ混入欠陥の検出し易さも向上させる
- ISO26262にはそれらの観点が記載されている（但し、基準値は無い）



■「効率的な設計」と「安全な設計」は同じではない
- コスト／納期／性能の前に、まず信頼性／安全性

事例：安全な設計のための設計原則

えっ！割込みを使うのは良くないの！？

■ソフトウェア設計ガイドラインから抜粋

（ソフトウェアアーキテクチャ設計の設計原則）

	項目	定義	ASIL A	ASIL B	ASIL C	ASIL D
a	階層構造		++	++	++	++
b	コンポーネントサイズ制限		++	++	++	++
c	インターフェースサイズ制限		+	+	+	+
d	高い凝集度		+	++	++	++
e	弱い結合度		+	++	++	++
f	適切なスケジューリング特性		++	++	++	++
g	割込みの使用制限	割込み処理の使用を制限し、ポーリングなどで実現する。 ・「制限」は、他の設計の検討とのバランスの中で最小化することを意味する。 ・いかなる割り込みも優先順位を元に使用されねばならない。	+	+	+	++

省略

■割込みの使用制限（解説）

定義	基準値
多重割込み	禁止
タイマ割込み数	1以下 カウンタ加減算、あるいはフラグ設定のみ
ハード割込み数	できるだけ使用しない
関連コンポーネント数	1データ毎に、送信／受信各1以下

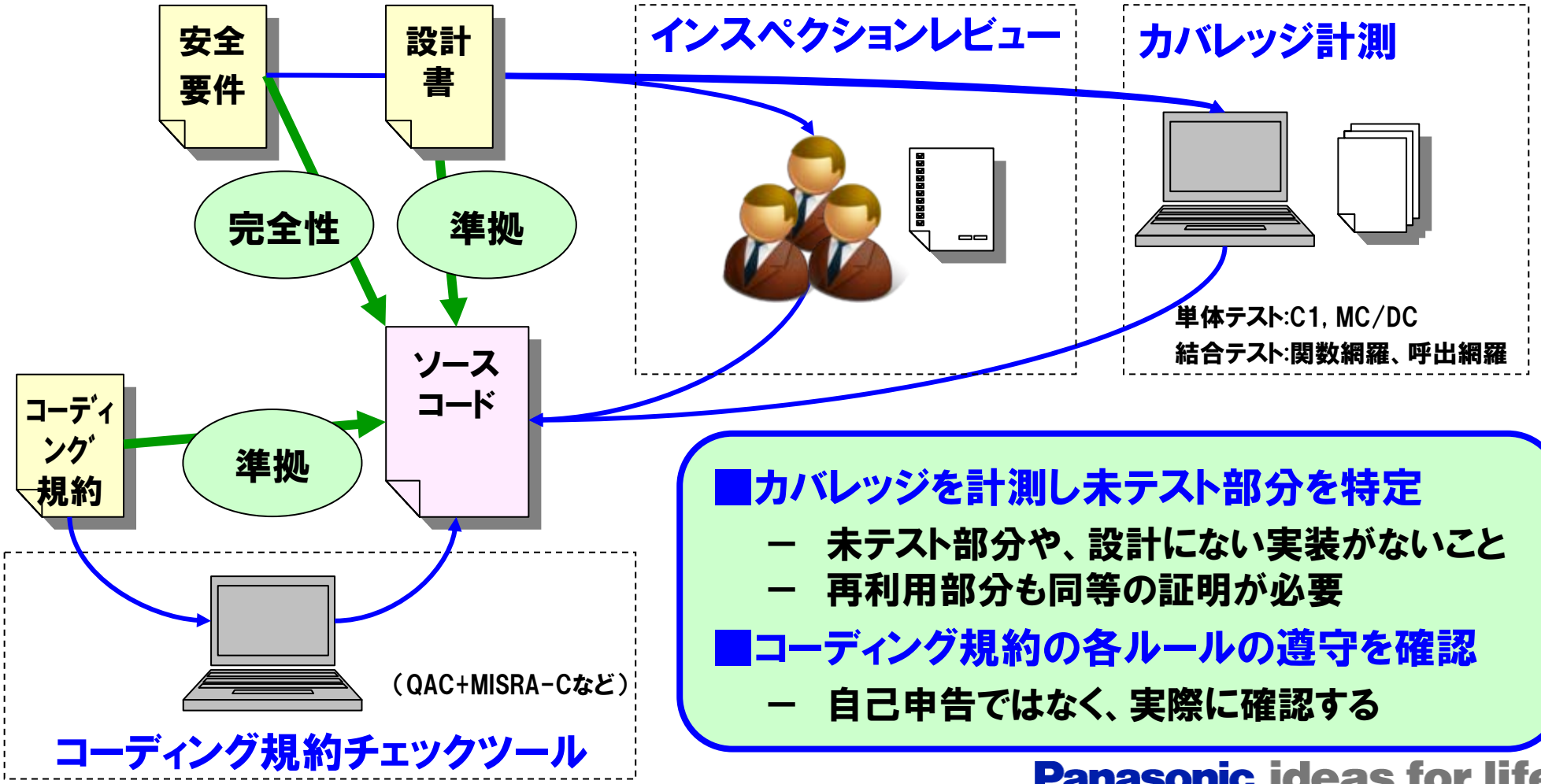
割込み処理を多用すると、ソフトウェアの動きを一意に定めることが困難になり、レビューやテストで検出困難な予想外の動きによる不具合の可能性が高まる。このため、特にASIL Dの場合は、割込みの使用は極力避け、ポーリングなどで対応するのが望ましい。
割込み処理を禁止しているわけではないが、上記の理由でOEM（カーメーカー）は割込みの使用を嫌う。

■基準値だけでなく、目的や根拠を明記する

－ どんな欠陥の混入を防止するためのものかを開発者に伝え、判断できるようにする

③一貫性、完全性、準拠性の証明

安全要件からの一貫性を確実にすることで、安全目標の達成を確実にする
－ トレーサビリティとツールや公式なレビューによる確認



■カバレッジを計測し未テスト部分を特定
- 未テスト部分や、設計にない実装がないこと
- 再利用部分も同等の証明が必要

■コーディング規約の各ルールの遵守を確認
- 自己申告ではなく、実際に確認する

事例：テストの完全性

カバレッジ100％は必須なの？ テスト件数率やバグ収束率では判断しない？

■ソフトウェアテストガイドラインから抜粋

	観点	確認方法		テスト環境	
		テスト手法	テストケース抽出手法	実機	擬似
1	ソフトウェアユニット設計仕様の準拠				
		省略			
2	ハードウェア-ソフトウェアインターフェース仕様の準拠				
3	機能が正しく動作している	要件ベーステスト	境界値の分析 誤り推測	○	△
		インターフェーステスト	境界値の分析 誤り推測		
		リソース使用テスト	要件の分析		
		モデルとコード間のバック-to-バックテスト	※モデルでのテストケース		
4	意図しない機能が無い	1）～3）の実施の結果をカバレッジ計測する		○	○
5	ロバスト性				
		省略			
6	機能をサポートするリソースの十分性				

テストの観点毎に手法を定義

各手法は開発者が実施可能なレベルに、詳細な解説を記述

ソフトウェアテストガイドラインから抜粋

■各テスト手法／テストケース抽出手法を実施することで、準拠性や完全性を確認する

- カバレッジ90%は、10%**未テスト部分**があると見なされる
- 人によってテストの質が違うと、テスト件数やバグ件数は品質の証明には使えない

■背景
・従来開発の課題

■車載系と家電系のソフトウェア開発との違い
・機能安全とは
・機能安全のための活動
－安全性
－良い設計
－確認

■まとめ

・ISO26262規格の車載系以外への応用の可能性

まとめ／今後の予定

■車載用ISO26262のノウハウは車載以外にも適用できる

- 手法や観点などは、ほぼそのまま使用できる
- 説明可能な信頼性／安全性を実現できる(はず)

- 家電向けにはテーラリング(取捨選択)が必要
- 適切な手法を正しく活用できるスキルが必要

今後の予定

- 車載系のプロジェクトでの試行中 (2010/7～10予定)
- 高信頼性を求められる一般家電向けのテーラリングと展開 (計画中)