http://www.exmotion.co.jp/

要求仕様定義USDMによる自動車システム開発の弱点の克服

～開発の空洞を埋め、上流から下流までをシームレスに繋ぐには？

株式会社エクスモーション　高橋久憲

2014/6/6

発表の流れ

- 会社紹介
- 自動車の制御開発について
- 制御開発における課題
- USDMによる暗黙知の形式化
- 暗黙知を形式化した効果
- まとめ

会社紹介

エクスモーションの事業スタイル

ソフトウェアの「品質」を『フロントローディング』に付加することにで、競争力ある製品を作り出します

お客さま（機能開発のプロ集団）

製品の競争力の源泉となる
「機能」「要素技術」に注力

・要素技術の発見・研磨
・現実世界の困難を克服する制御技術など

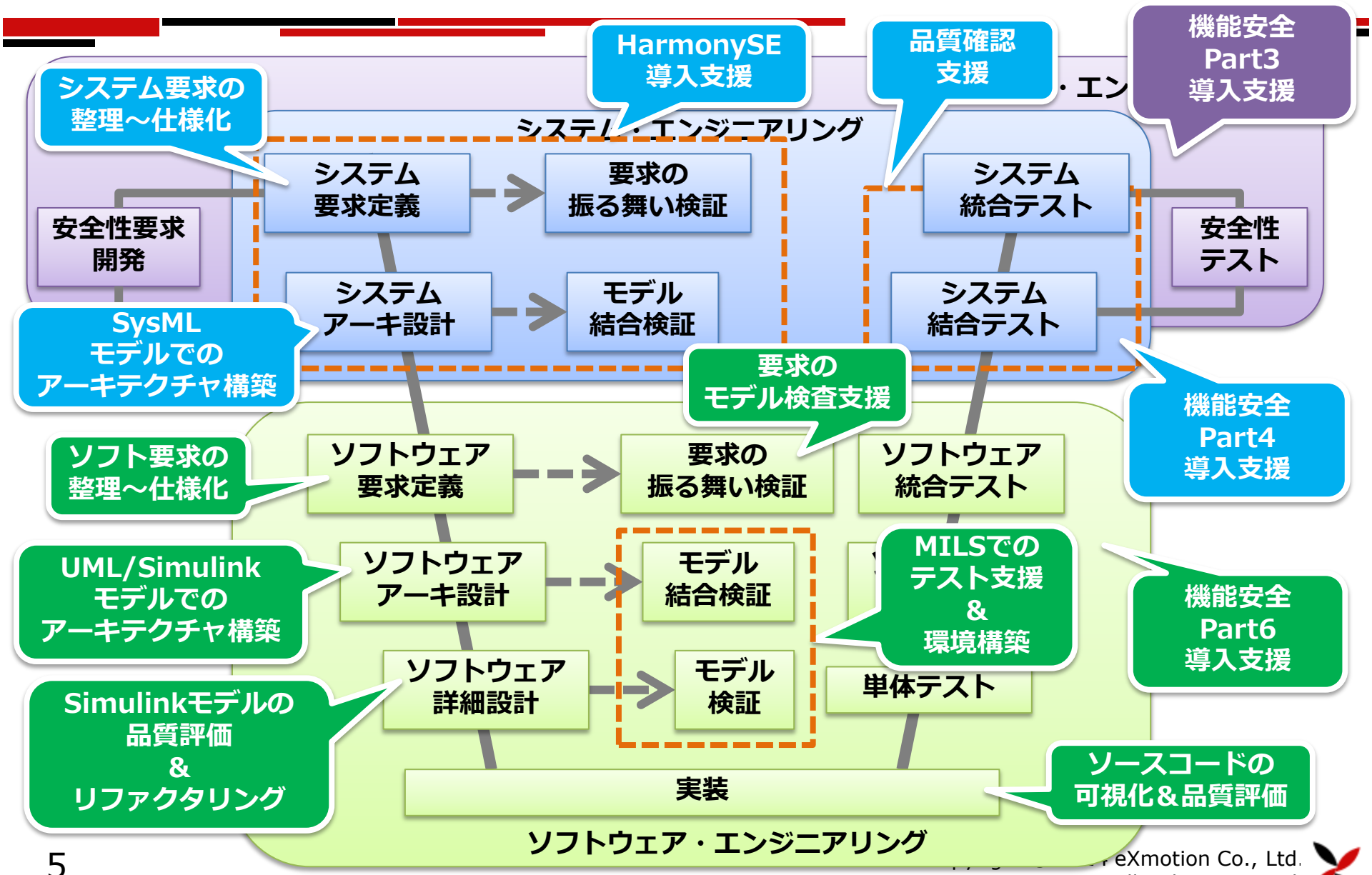
市場で闘える製品

エクスモーション（設計技術のプロ集団）

「機能」「要素技術」を
製品に反映するための
「アーキテクチャ」「開発プロセス」
に注力

・要求整理から設計・テストまでのトレーサビリティ確保
・変更容易性や信頼性の向上

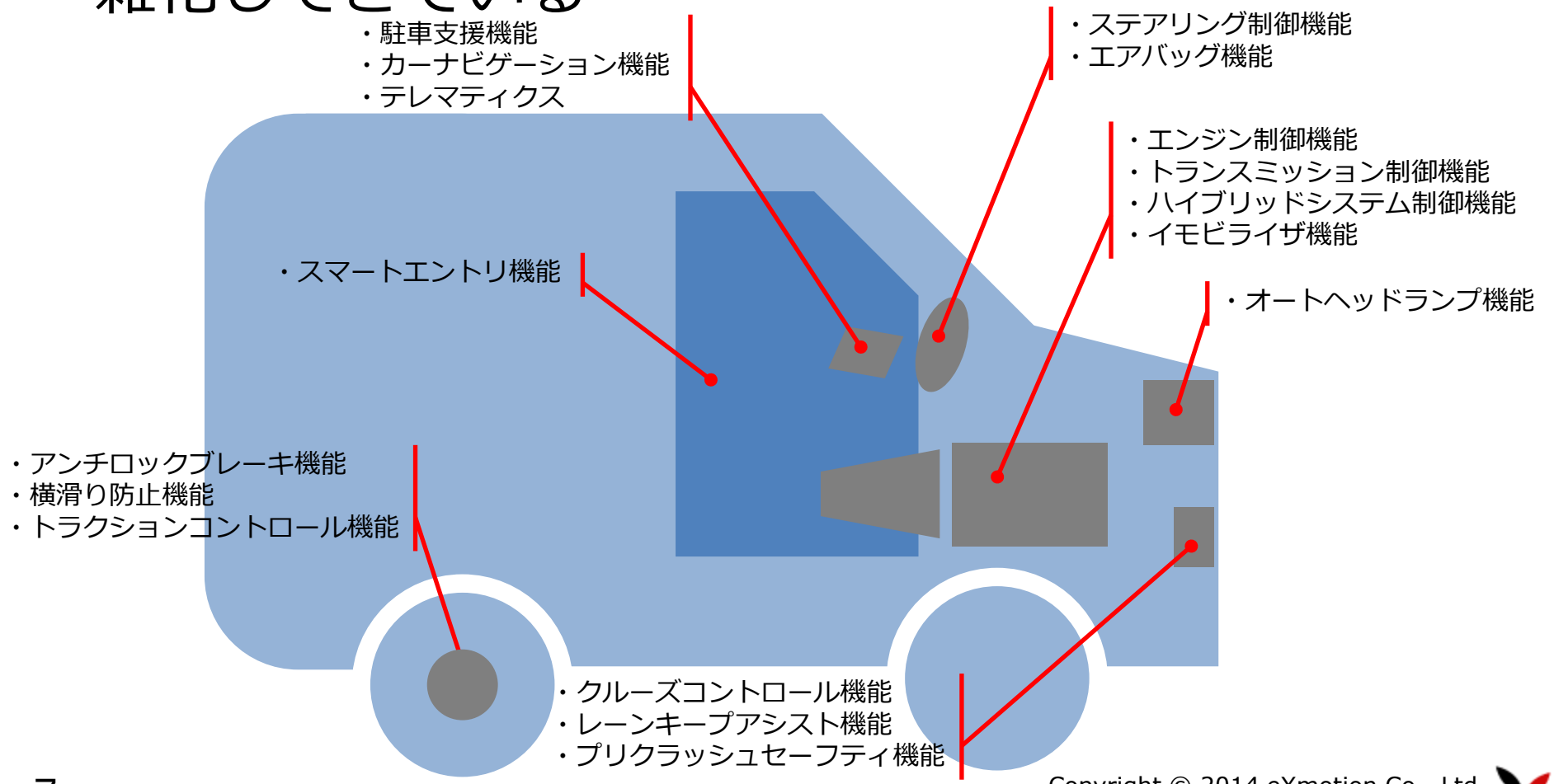
自動車分野での豊富な実績



自動車の制御開発について

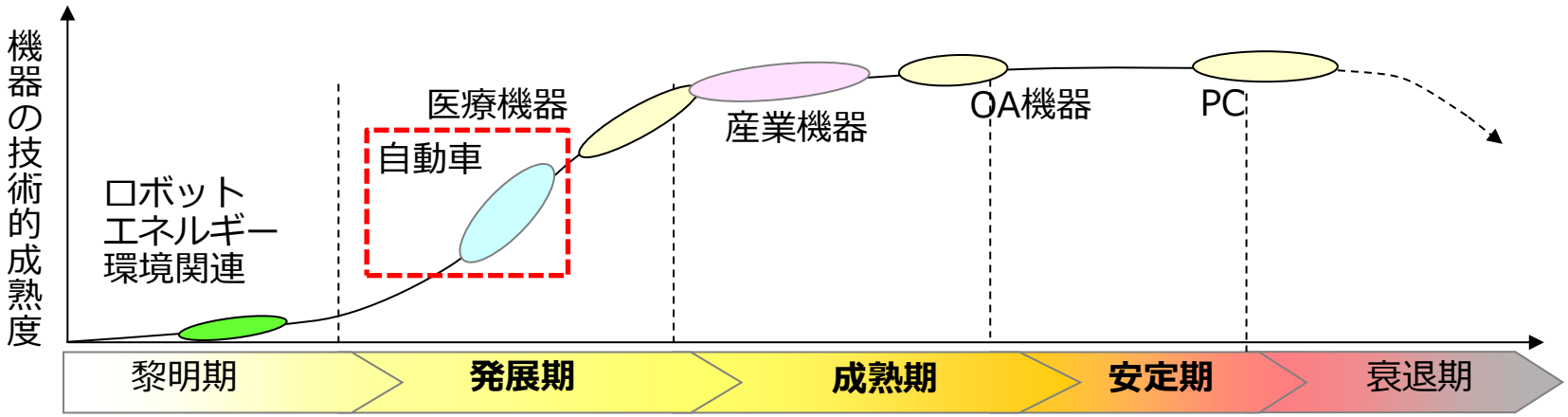
自動車は複雑化している

- 近年、自動車にはたくさんの機能が追加され、複雑化してきている



自動車は複雑化している

- それは自動車システムが技術的な発展期だから



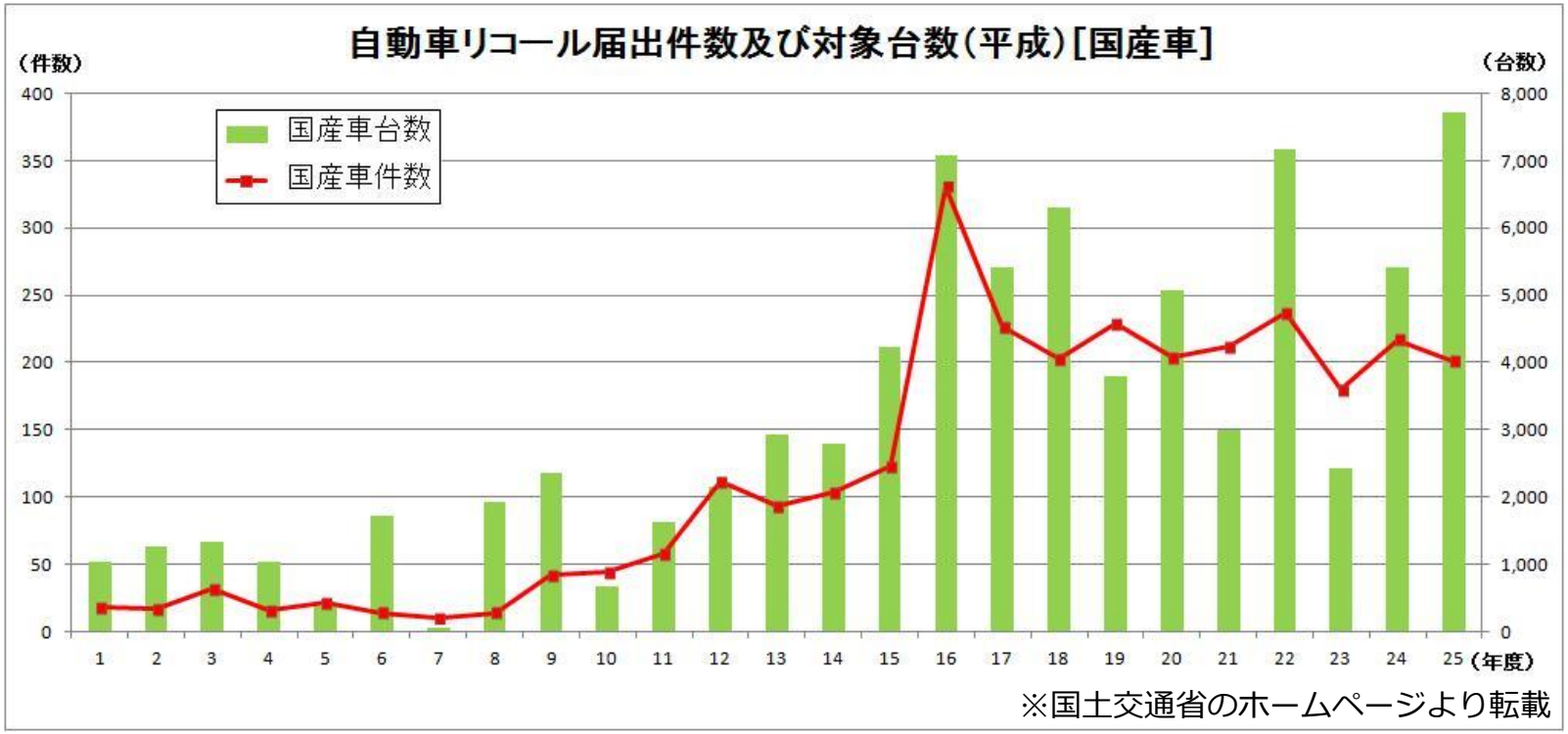
ソリューションフェア2009　(c) Mitsuyuki Hoshi
の資料を基に加筆、変更して使用

発展期は、競争のために新機能を
・たくさん
・他社よりも早く
追加することが必要

短期間で効率よく開発することが求められる

不具合も増加している

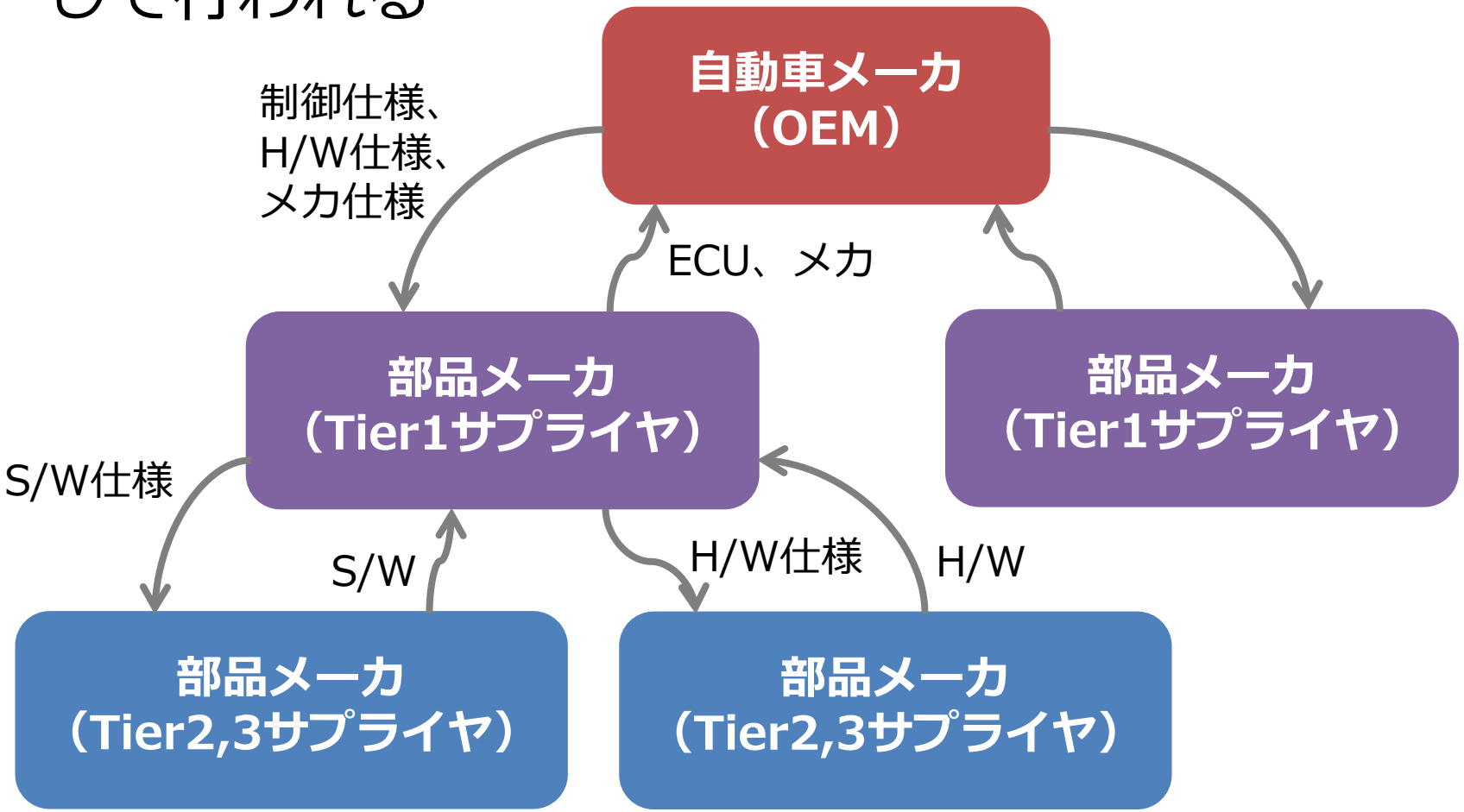
- 一方、自動車の不具合もこの10年で増加している



不具合件数の増加の原因のひとつに、
自動車システムが複雑化していることが考えられる

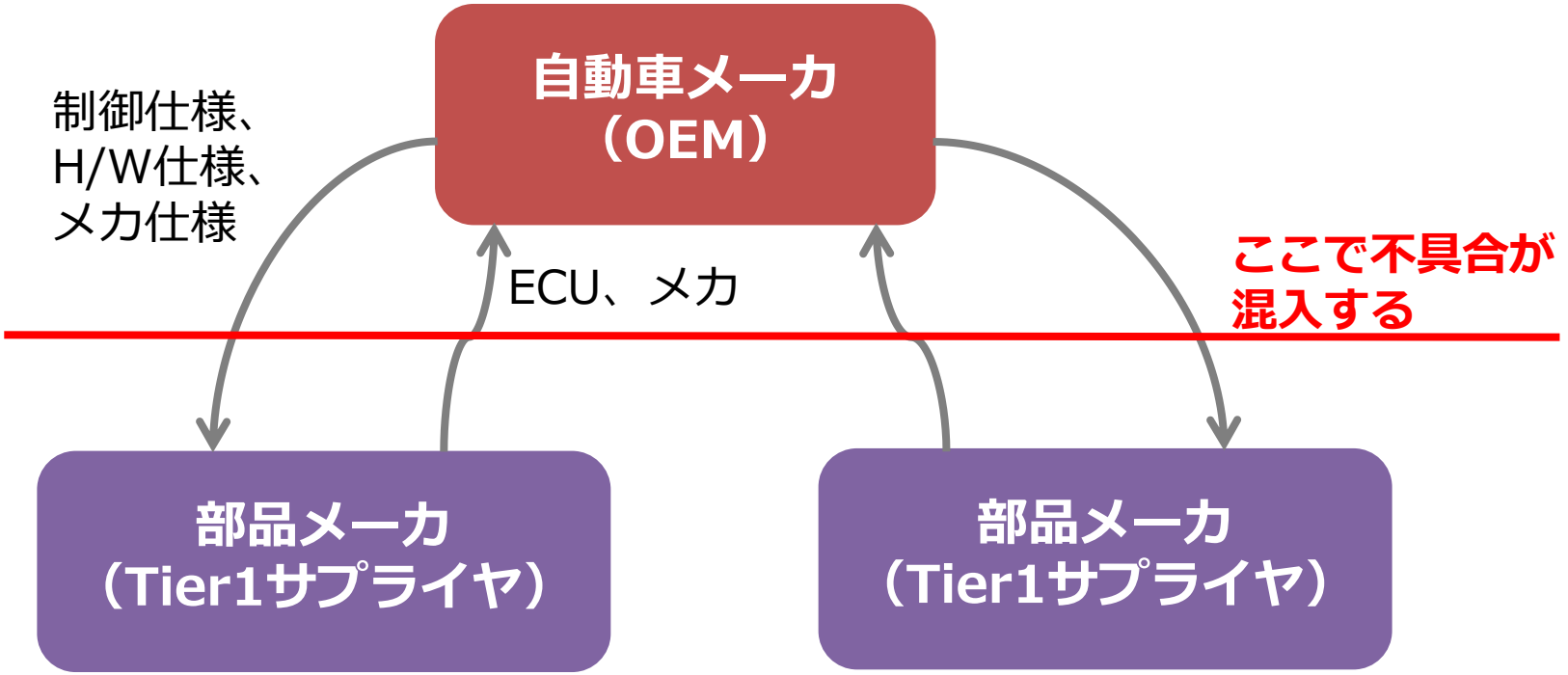
自動車開発の体制

- 自動車の開発は自動車メーカと部品メーカが連携して行われる



不具合混入の原因

- システムが複雑化するなかで、
  自動車メーカと部品メーカの仕様に関する認識が一致していない
  ことに起因する不具合も増えていると考えられる



日本の自動車産業が優位に立ち続けるには

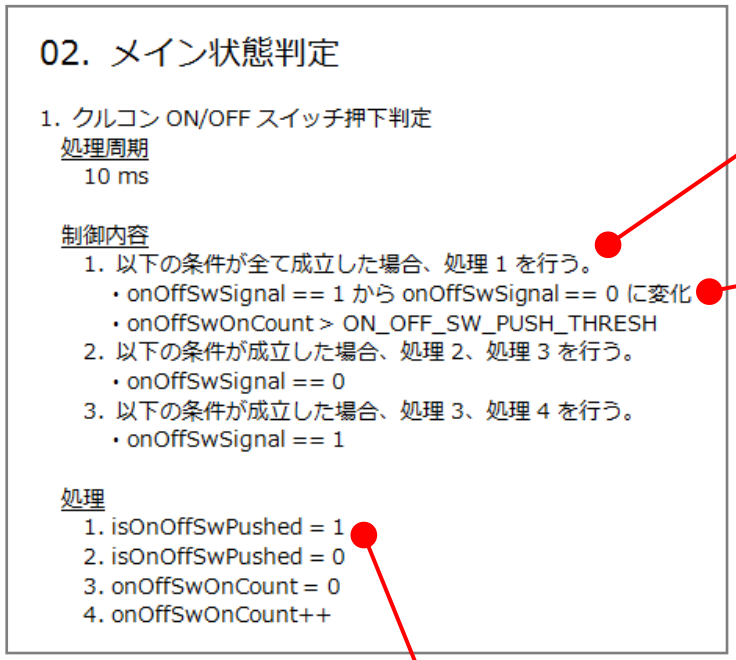
- 今後も日本の自動車産業が世界の市場で優位に立ち続けるには、魅力的な機能を早く搭載しながらも、高い品質を維持することが重要

- そのためには、不具合の混入原因のひとつである、**自動車メーカと部品メーカの仕様に関する認識が一致していない**という点を改善する必要がある

制御開発における課題

仕様に関する認識が一致しない原因

- 自動車システムの制御ソフトウェア開発の場合、自動車メーカから部品メーカに“どのように制御するか”を説明した仕様書がインプットされる



02. メイン状態判定

1. クルコン ON/OFF スイッチ押下判定

処理周期

10 ms

制御内容

1. 以下の条件が全て成立した場合、処理 1 を行う。
   - onOffSwSignal == 1 から onOffSwSignal == 0 に変化
   - onOffSwOnCount > ON_OFF_SW_PUSH_THRESH
2. 以下の条件が成立した場合、処理 2、処理 3 を行う。
   - onOffSwSignal == 0
3. 以下の条件が成立した場合、処理 3、処理 4 を行う。
   - onOffSwSignal == 1

処理

1. isOnOffSwPushed = 1
2. isOnOffSwPushed = 0
3. onOffSwOnCount = 0
4. onOffSwOnCount++

この処理が“なぜ”必要なのか書かれていない

この条件が“何を”意味しているかは書かれていない

ここで行っていることが“何か”は書かれていない

この成果物には実現手段である“仕様”だけが記載されている

⇨ **要求が暗黙知となっている**

要求が暗黙知になることの問題点

- 自動車メーカでは
  - 開発を担当した人だけがその制御に関する要求を把握している状態になってしまい、その人がいなくなった時に知識が失われる
  - 機能を変更するときに、他に関係のある機能が分からず、影響が分析できない
- 部品メーカでは
  - 要求が分からないまま、仕様に従って実装することで、仕様に矛盾があっても気づかない
  - 仕様では記載されてないが、設計上判断が必要な場合に、誤った判断をしてしまう

暗黙知を形式知にするには

- これらの問題点を解決するには、制御仕様に対する要求を把握し、それを自動車メーカの開発者、部品メーカの開発者どちらにも分かりやすい形式で文書化する必要がある

制御仕様書

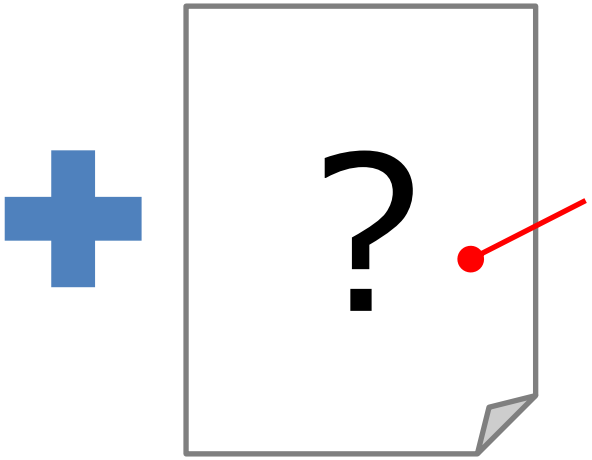
02. メイン状態判定

1. クルコン ON/OFF スイッチ押下判定
   処理周期
   10 ms

   制御内容
   1. 以下の条件が全て成立した場合、処理 1 を行う。
      ・onOffSwSignal == 1 から onOffSwSignal == 0 に変化
      ・onOffSwOnCount > ON_OFF_SW_PUSH_THRESH
   2. 以下の条件が成立した場合、処理 2、処理 3 を行う。
      ・onOffSwSignal == 0
   3. 以下の条件が成立した場合、処理 3、処理 4 を行う。
      ・onOffSwSignal == 1

   処理
   1. isOnOffSwPushed = 1
   2. isOnOffSwPushed = 0
   3. onOffSwOnCount = 0
   4. onOffSwOnCount++

制御の要求を記載した文書



どのようなものが良いか？

USDMによる暗黙知の形式化

分かりやすい要求仕様にするには

- すでに制御の流れを把握している開発者であれば、制御仕様書に要求を追記するだけで理解できる
- しかし、制御仕様の詳細を理解していない人が、そのシステムの振る舞いを把握したい場合がある
- そのような人にシステムを理解してもらうためには、要求仕様として以下のような条件を満たす必要がある
  - ① 必要な情報がもれなく記載されている
  - ② 概要→詳細と段階を追って理解することができる
  - ③ 処理の流れを理解することができる
  - ④ 可読性が高い

要求記述方法の比較

要求記述方法	メリット	デメリット
箇条書き	教育しなくても書き始められる	要求の粒度がそろえにくい
ユースケース	処理の流れが分かりやすい-③ 可読性は高い-④	導入に教育が必要 その機能が必要な理由が表現しにくい-① 仕様が表現できない-②
SysMLの要求図	要求間の複雑な関係を表現できる	導入に教育が必要 詳細を記載するとモデルが巨大化する-④
USDM	粒度をそろえやすい 理由の記載を強制できる-① 仕様が表現できる-② 処理の流れが分かりやすい-③ 詳細を表現しても可読性が悪くなりにくい-④	導入に教育が必要 要求間に複雑な関係があった場合は表現しにくい

USDMであれば、読みやすい要求仕様の条件を満たせる

USDMによって制御の要求仕様を定義

- 対象システムの制御仕様から機能要求を抽出し USDMの形式で文書化することが有効である

制御仕様書

02. メイン状態判定

1. クルコン ON/OFF スイッチ押下判定

処理周期
10 ms

制御内容
1. 以下の条件が全て成立した場合、処理 1 を行う。
   - onOffSwSignal == 1 から onOffSwSignal == 0 に変化
   - onOffSwOnCount > ON_OFF_SW_PUSH_THRESH
2. 以下の条件が成立した場合、処理 2、処理 3 を行う。
   - onOffSwSignal == 0
3. 以下の条件が成立した場合、処理 3、処理 4 を行う。
   - onOffSwSignal == 1

処理
1. isOnC
2. isOnC
3. onOff
4. onOff

03. ACC セット状態判定

1. メイン状態 OFF 判定

処理周期
10 ms

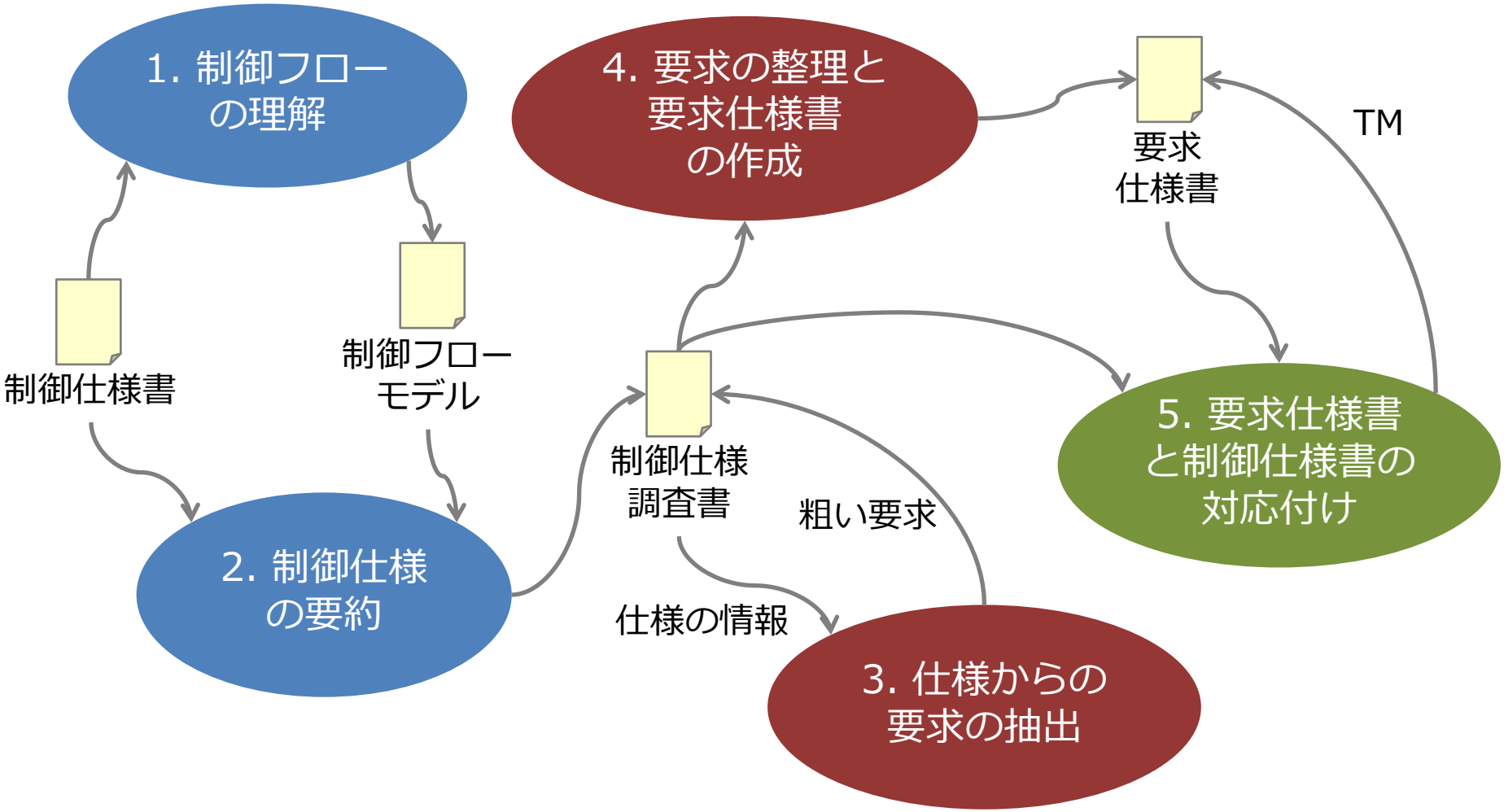
制御内容
1. 以下の条件が成立している場合、処理 1 を行う。
   - accMainState == ACC_MAIN_OFF
2. それ以外は処理 2 を行う。

処理
1. isAccMainOff = 1
2. isAccMainOff = 0

要求仕様書

<クルコンの始動>			
サブシステム要求	ACC.01.01		クルコンがOFFの時にドライバがON/OFFスイッチを押下した場合、クルコンをONできる条件が成立していることを判定し、クルコンをONにする。
	理由		クルコンを使わない状況では、動作しないように機能をOFFできるようにする必要がある。
	説明		特になし。
	<ON/OFFスイッチの押下判定>		
	ソフト要求	ACC.01.01.01	スイッチのチャタリングの影響を考慮し、ドライバがON/OFFスイッチを押下したことを確実に検出する。
		理由	チャタリングによって誤判定しないようにするため。
		説明	特になし。
		<押下判定>	
		SP.ACC.001.01	ON/OFFスイッチの信号が「オン」である状態が50[ms]以上継続した後に「オフ」に変化した場合、ON/OFFスイッチ押下判定を「オン」にする。
		SP.ACC.001.02	ON/OFFスイッチの信号が「オフ」であるか、「オン」が50[ms]以上継続しないで「オフ」に変化した場合は、ON/OFFスイッチ押下判定を「オフ」にする。
	<クルコンON/OFF状態の遷移>		
	ソフト要求	ACC.01.01.02	クルコンON/OFF状態を保持し、ON/OFFスイッチが押下されたときにクルコンをONにできる条件が成立している場合は、クルコンON/OFF状態をOFFからONに遷移させる。
		理由	クルコンのON/OFFの状態を管理するため。
		説明	特になし。
		<状態の保持>	
		SP.ACC.002.21	以下の条件が全て成立している場合、メインONを許可する。 ・イグニッションが「オン」である ・診断の結果、システムに異常が発生していない いずれかが成立していない場合は許可しない。

要求仕様作成の流れ



1. 制御フローの理解

- 制御仕様に基づき制御フローをモデル化することで制御の前後関係、情報の入出力関係を理解する

制御仕様書

02. メイン状態判定

1. クルコン ON/OFF スイッチ押下判定
   処理周期
   10 ms

   制御内容
   1. 以下の条件が全て成立した場合、処理 1 を行う。
      ・onOffSwSignal == 1 から onOffSwSignal == 0 に変化
      ・onOffSwOnCount > ON_OFF_SW_PUSH_THRESH
   2. 以下の条件が成立した場合、処理 2、処理 3 を行う。
      ・onOffSwSignal == 0
   3. 以下の条件が成立した場合、処理 3、処理 4 を行う。
      ・onOffSwSignal == 1

   処理
   1. isOnOffSwPushed = 1
   2. isOnOffSw
   3. onOffSwC
   4. onOffSwC

流れが
把握しにくい

03. ACC セット状態判定

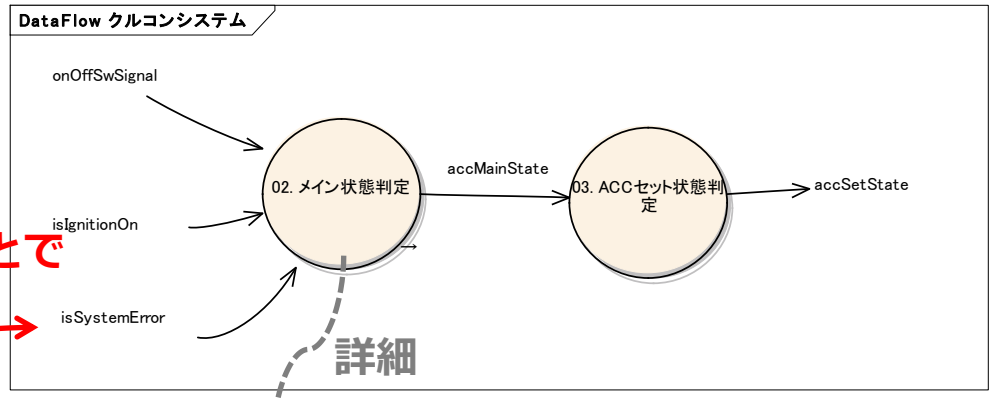
1. メイン状態 OFF 判定
   処理周期
   10 ms

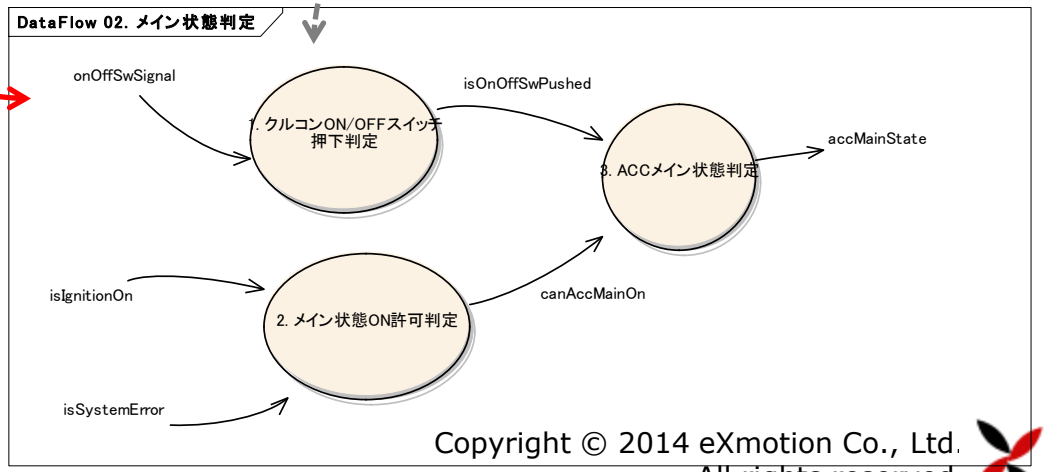
   制御内容
   1. 以下の条件が成立している場合、処理 1 を行う。
      ・accMainState == ACC_MAIN_OFF
   2. それ以外は処理 2 を行う。

   処理
   1. isAccMainOff = 1
   2. isAccMainOff = 0

モデル化することで
流れを把握

制御フローモデル





2. 制御仕様の要約

- 制御仕様書の処理内容を理解し、そこで何を行っているかを整理する

制御仕様書

02. メイン状態判定

1. クルコン ON/OFF スイッチ押下判定

処理周期
10 ms

制御内容
1. 以下の条件が全て成立した場合、処理 1 を行う。
   - onOffSwSignal == 1 から onOffSwSignal == 0 に変化
   - onOffSwOnCount > ON_OFF_SW_PUSH_THRESH
2. 以下の条件が成立した場合、処理 2、処理 3 を行う。
   - onOffSwSignal == 0
3. 以下の条件が成立した場合、処理 3、処理 4 を行う。
   - onOffSwSignal == 1

処理
1. isOnOffSwPushed = 1
2. isOnOffSwPushed = 0
3. onOffSwOnCount = 0
4. onOffSwOnCount++

処理内容を整理

制御仕様調査書

02. メイン状態判定		
概要	ON/OFFスイッチの操作から、ACCのメイン状態を切り替える。	
章番号	章タイトル	制御仕様内容
1	クルコンON/OFFスイッチ押下判定	ON/OFFスイッチの信号が「オン」である状態が50[ms]以上継続した後に「オフ」に変化した場合、ON/OFFスイッチ押下判定を「オン」にする。
		ON/OFFスイッチの信号が「オフ」であるか、「オン」が50[ms]以上継続しないで「オフ」に変化した場合は、ON/OFFスイッチ押下判定を「オフ」にする。
2	メイン状態ON許可判定	以下の条件が全て成立している場合、メインONを許可する。 ・イグニッションが「オン」である ・診断の結果、システムに異常が発生していない いずれかが成立していない場合は許可しない。

3. 仕様からの要求の抽出

- 整理した仕様から「その処理は何をするために行っているか？」を考えて要求を抽出する
- 要求が不明な場合は、開発者に対してヒアリングを行う

制御仕様調査書

02. メイン状態判定

概要	ON/OFFスイッチの操作から、ACCのメイン状態を切り替える。

章番号	章タイトル	制御仕様内容	要求
1	クルコンON/OFFスイッチ押下判定	ON/OFFスイッチの信号が「オン」である状態が50[ms]以上継続した後に「オフ」に変化した場合、ON/OFFスイッチ押下判定を「オン」にする。	スイッチのチャタリングの影響を考慮し、ドライバがON/OFFスイッチを押下したことを確実に検出する。
		ON/OFFスイッチの信号が「オフ」であるか、「オン」が50[ms]以上継続しないで「オフ」に変化した場合は、ON/OFFスイッチ押下判定を「オフ」にする。	

要求を抽出

4. 仕様の整理と要求仕様書の作成

- 抽出した要求、仕様を時系列な処理の流れが分かるように整理して要求仕様にまとめる

制御仕様調査書

02. メイン状態判定

概要	ON/OFFスイッチの操作から、ACCのメイン状態を切り替える。

章番号	章タイトル	制御仕様内容	要求
1	クルコンON/OFFスイッチ押下判定	ON/OFFスイッチの信号が「オン」である状態が50[ms]以上継続した後に「オフ」に変化した場合、ON/OFFスイッチ押下判定を「オン」にする。	スイッチのチャタリングの影響を考慮し、ドライバがON/OFFスイッチを押下したことを確実に検出する。
		ON/OFFスイッチの信号が「オフ」であるか、「オン」が50[ms]以上継続しないで「オフ」に変化した場合は、ON/OFFスイッチ押下判定を「オフ」にする。	

要求を記載

仕様を記載

要求仕様書

サブシステム要求	ACC.01.01	クルコンがOFFの時にドライバがON/OFFスイッチを押下した場合、クルコンをONできる条件が成立していることを判定し、クルコンをONにする。	
	理由	クルコンを使わない状況では、動作しないように機能をOFFできるようにする必要がある。	
	説明	特になし。	
	<ON/OFFスイッチの押下判定>		
ソフト要求	ACC.01.01.01	スイッチのチャタリングの影響を考慮し、ドライバがON/OFFスイッチを押下したことを確実に検出する。	
	理由	チャタリングによって誤判定しないようにするため。	
	説明	特になし。	
	<押下判定>		
	SP.ACC.001.01	ON/OFFスイッチの信号が「オン」である状態が50[ms]以上継続した後に「オフ」に変化した場合、ON/OFFスイッチ押下判定を「オン」にする。	
	SP.ACC.001.02	ON/OFFスイッチの信号が「オフ」であるか、「オン」が50[ms]以上継続しないで「オフ」に変化した場合は、ON/OFFスイッチ押下判定を「オフ」にする。	

5. 要求仕様と制御仕様の対応付け

- TMを付け制御仕様書と対応付けることで、全ての制御仕様書から要求を抽出したことを確認する

要求仕様書

クルコンECU			要求と仕様		制御仕様書 1.入力処理	2.メイン状態判定	3.ACCセット状態判定	4.目標車速決定	5.目標トルク算出
	<CANデータの受信>								
	<クルコンの始動>								
クルコンON	サブシステム要求	ACC.01.01	クルコンがOFFの時にドライバがON/OFFスイッチを押下した場合、クルコンをONできる条件が成立していることを判定し、クルコンをONにする。						
		理由	クルコンを使わない状況では、動作しないように機能をOFFできるようにする必要がある。						
		説明	特になし。						
		<ON/OFFスイッチの押下判定>							
		ソフト要求	ACC.01.01.01	スイッチのチャタリングの影響を考慮し、ドライバがON/OFFスイッチを押下したことを確実に検出する。					
			理由	チャタリングによって誤判定しないようにするため。					
			説明	特になし。					
			<押下判定>						
			SP.ACC.001.01	ON/OFFスイッチの信号が「オン」である状態が50[ms]以上継続した後に「オフ」に変化した場合、ON/OFFスイッチ押下判定を「オン」にする。		1			
			SP.ACC.001.02	ON/OFFスイッチの信号が「オフ」であるか、「オン」が50[ms]以上継続しないで「オフ」に変化した場合は、ON/OFFスイッチ押下判定を「オフ」にする。		1			

TM

要求仕様書の完成

- このように、ボトムアップのアプローチで要求仕様書を作成することができる

要求だけを読めば、
どのような流れで
何をしているかが把握できる

どの制御仕様書で
どの処理が行われているか
把握できる

その処理が必要な
理由も把握できる

					制御仕様書				
					1.入力処理	2.メイン状態判定	3.ACCセット状態判定	4.目標車速決定	5.目標トルク算出
	<CANデータの受信>								
	<クルコンの始動>								
クルコンON	サブシステム要求	ACC.01.01		クルコンがOFFの時にドライバがON/OFFスイッチを押下した場合、クルコンをONできる条件が成立していることを判定し、クルコンをONにする。					
		理由		クルコンを使わない状況では、動作しないように機能をOFFできるようにする必要がある。					
		説明		特になし。					
		<ON/OFFスイッチの押下判定>							
		ソフト要求	ACC.01.01.01	スイッチのチャタリングの影響を考慮し、ドライバがON/OFFスイッチを押下したことを確実に検出する。					
			理由	チャタリングによって誤判定しないようにするため。					
			説明	特になし。					
			<押下判定>						
			SP.ACC.001.01	ON/OFFスイッチの信号が「オン」である状態が50[ms]以上継続した後に「オフ」に変化した場合、ON/OFFスイッチ押下判定を「オン」にする。		1			
			SP.ACC.001.02	ON/OFFスイッチの信号が「オフ」であるか、「オン」が50[ms]以上継続しないで「オフ」に変化した場合は、ON/OFFスイッチ押下判定を「オフ」にする。		1			

暗黙知を形式化した効果

5. 要求仕様と制御仕様の対応付け

- 要求仕様を作成することによって、自動車メーカと部品メーカの連携ミスによる不具合がどの程度解消できたか、定量的な計測はできていない
- ただ、実務のレベルでは以下のような効果が認められた
  - ① 仕様の不備を検出できる
  - ② 受託側の誤解リスクを回避できる
  - ③ テストの質が向上する
  - ④ 自動車向け機能安全規格への対応がしやすくなる
  - ⑤ アーキテクチャ設計の質が向上する
  - ⑥ システム設計の質が向上する

①仕様の不備を検出できる

- 要求が書かれていないため、制御仕様だけを見てレビューをしても、第三者が要求に対する仕様の妥当性（Validation）を確認することは難しい

02. メイン状態判定

1. クルコン ON/OFF スイッチ押下判定

処理周期

10 ms

制御内容

1. 以下の条件が全て成立した場合、処理 1 を行う。
   - onOffSwSignal == 1 から onOffSwSignal == 0 に変化
   - onOffSwOnCount > ON_OFF_SW_PUSH_THRESH
2. 以下の条件が成立した場合、処理 2、処理 3 を行う。
   - onOffSwSignal == 0
3. 以下の条件が成立した場合、処理 3、処理 4 を行う。
   - onOffSwSignal == 1

処理

1. isOnOffSwPushed = 1
2. isOnOffSwPushed = 0
3. onOffSwOnCount = 0
4. onOffSwOnCount++

この処理が“なぜ”必要なのか書かれていない

要求に対する仕様の妥当性を判断するのに必要な情報がない

①仕様の不備を検出できる

- 要求仕様でレビューすることで、第三者が要求と見比べながら制御仕様の妥当性を確認できる

<クルコンの始動>			
サブシステム要求	ACC.01.01	クルコンがOFFの時にドライバがON/OFFスイッチを押下した場合、クルコンをONできる条件が成立していることを判定し、クルコンをONにする。	
	理由	クルコンを使わない状況では、動作しないように機能をOFFできるようにする必要がある。	
	説明	特になし。	
	<ON/OFFスイッチの押下判定>		
	ソフト要求	ACC.01.01.01	スイッチのチャタリングの影響を考慮し、ドライバがON/OFFスイッチを押下したことを確実に検出する。
		理由	チャタリングによる誤判定しないようにするため。
		説明	特になし。
		<押下判定>	
		SP.ACC.001.01	ON/OFFスイッチの信号が「オン」である状態が50[ms]以上継続した後に「オフ」に変化した場合、ON/OFFスイッチ押下判定を「オン」にする。
		SP.ACC.001.02	ON/OFFスイッチの信号が「オフ」であるか、「オン」が50[ms]以上継続しないで「オフ」に変化した場合は、ON/OFFスイッチ押下判定を「オフ」にする。

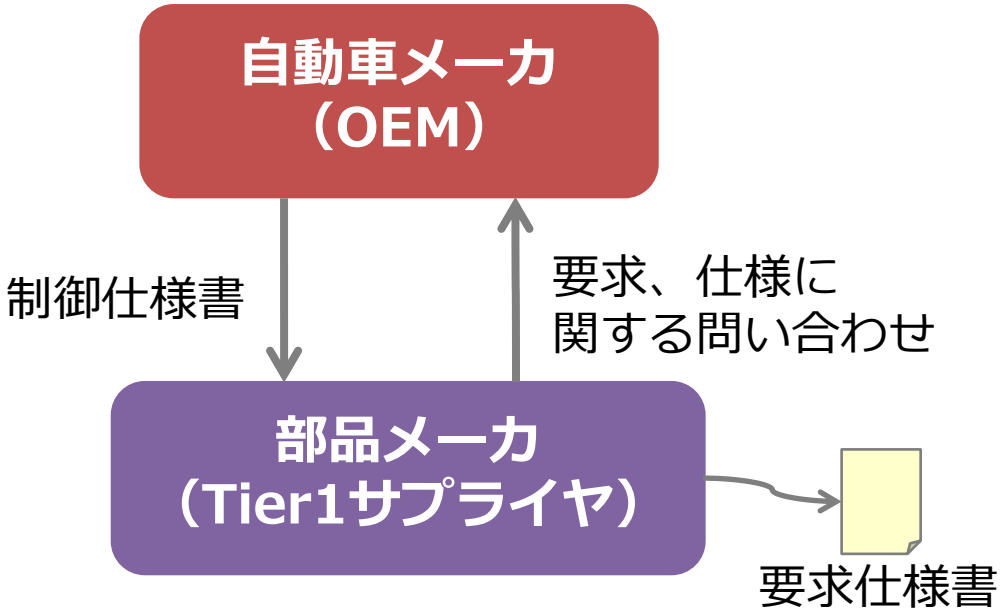
チャタリングへの対応はOK

「オフ」しないと押下と判定しないが、長押しした時に「オン」と判定できなくて良い？

- レビュー効率が向上するため、少ない工数で十分な検証を行うことができる

②受託側の誤解リスクを回避できる

- 部品メーカで、入力された制御仕様を基に要求を文書化するという運用も可能



**問い合わせを
要求仕様書ベースで
行うことにより、
自動車メーカと部品メーカの
認識を一致させられる**

- それにより、部品メーカが仕様の意図を誤解することで実装時に混入する不具合を防止できる

③テストの質が向上する

- 要求仕様書を作成していれば、要求仕様に基づいて網羅的にテストケースを抽出することができる

要求仕様書

<クルコンの始動>			
サブシステム要求	ACC.01.01	クルコンがOFFの時にドライバがON/OFFスイッチを押下した場合、クルコンをONできる条件が成立していることを判定し、クルコンをONにする。	
	理由	クルコンを使わない状況では、動作しないように機能をOFFできるようにする必要がある。	
	説明	特になし。	
	<ON/OFFスイッチの押下判定>		
	ソフト要求	ACC.01.01.01	スイッチのチャタリングの影響を考慮し、ドライバがON/OFFスイッチを押下したことを確実に検出する。
		理由	チャタリングによって誤判定しないようにするため。
		説明	特になし。
		<押下判定>	
		SP.ACC.001.01	ON/OFFスイッチの信号が「オン」である状態が50[ms]以上継続した後に「オフ」に変化した場合、ON/OFFスイッチ押下判定を「オン」にする。
		SP.ACC.001.02	ON/OFFスイッチの信号が「オフ」であるか、「オン」が50[ms]以上継続しないで「オフ」に変化した場合は、ON/OFFスイッチ押下判定を「オフ」にする。

テスト仕様書

	テスト項目	テストケース	ソフトウェア要求仕様	時間(ms)	入力値					
					onOffSwSignal	modeSwSignal	resAclSwSignal	setCoastSwSignal	cancelSwSignal	distanceSwSignal
					⌐	⌐	⌐	⌐	⌐	⌐
00-02	<ON/OFF スイッチの押下判定> スイッチをプッシュしたまま、50 ms 以上経過すれば、スイッチをリリースしたタイミングで、「ON/OFF押下判定」を「オン」にする。	スイッチを押下	SP.ACC.001.01	10	1	0	0	0	0	0
00-03		スイッチを押下したまま 60ms 経過 ⇒その間、「押下判定結果」はオフ		20	1	0	0	0	0	0
00-04				70	1	0	0	0	0	0
00-05		スイッチをリリース ⇒「押下判定結果」がオン		80	0	0	0	0	0	0
00-06		リリース後、10ms 経過 ⇒「押下判定結果」はオフ		90	0	0	0	0	0	0
00-07	スイッチをプッシュして 50 ms 以内にリリースした場合には「ON/OFF押下判定」は「オフ」のまま。	スイッチを押下	SP.ACC.001.02	100	1	0	0	0	0	0
00-08		スイッチを押下したまま30 ms 経過 ⇒その間、「押下判定結果」はオフ		110	1	0	0	0	0	0
00-09				130	1	0	0	0	0	0
00-10		スイッチをリリース ⇒「押下判定結果」はオフのまま		140	0	0	0	0	0	0
00-11		リリース後、10ms 経過 ⇒「押下判定結果」はオフのまま		150	0	0	0	0	0	0

要求仕様と対応付けてテストケースを作成・管理することで、テスト漏れを防ぐことができる

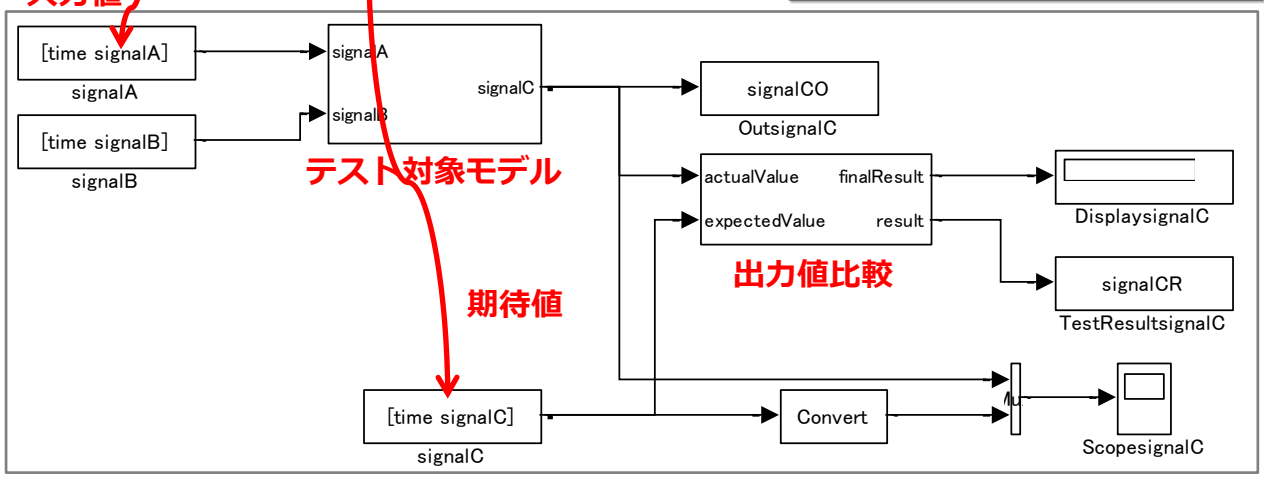
③テストの質が向上する

- モデルベース開発の場合、そのテスト仕様によってコード実装前に仕様の検証が可能となる

テスト仕様書

番号	テスト項目	テストケース	ソフトウェア要求仕様	時間(ms)	入力値 onOffSwSignal	modeSwSignal	resAdSwSignal	setCoastSwSignal	cancelSwSignal	distanceSwSignal	期待値 isResumeOperated	isSetOperated	isAccelerateOperated	isCoastOperated	isOnOffOperated	isModeOperated	isDistanceOperated	isCancelOperated
					⌐¬	⌐¬	⌐¬	⌐¬	⌐¬	⌐¬	⌐¬	⌐¬	⌐¬	⌐¬	⌐¬	⌐¬	⌐¬	⌐¬
					チェック要	チェック要	チェック要	チェック要	チェック要	チェック要	チェック要	チェック要	チェック要	チェック要	チェック要	チェック要	チェック要	チェック要
					0	0	0	0	0	0	0	0	0	0	0	0	0	0
					0	0	0	0	0	0	0	0	0	0	0	0	0	0
00-02		スイッチを押下		10	1	0	0	0	0	0	0	0	0	0	0	0	0	0
00-03	<ON/OFF スイッチの押下判定>	スイッチを押下したまま 60ms 経過 ⇒その間、「押下判定結果」はオフ		20	1	0	0	0	0	0	0	0	0	0	0	0	0	0
00-04	スイッチをプッシュしたまま、50 ms 以上経過すれば、スイッチをリリースしたタイミングで、「ON/OFF押下判定」を「オン」にする。		SP.ACC.001.01	70	1	0	0	0	0	0	0	0	0	0	0	0	0	0
00-05		スイッチをリリース ⇒「押下判定結果」がオン		80	0	0	0	0	0	0	0	0	0	0	1	0	0	0
00-06		リリース後、10ms 経過 ⇒「押下判定結果」はオフ		90	0	0	0	0	0	0	0	0	0	0	0	0	0	0

テスト用Simulinkモデル



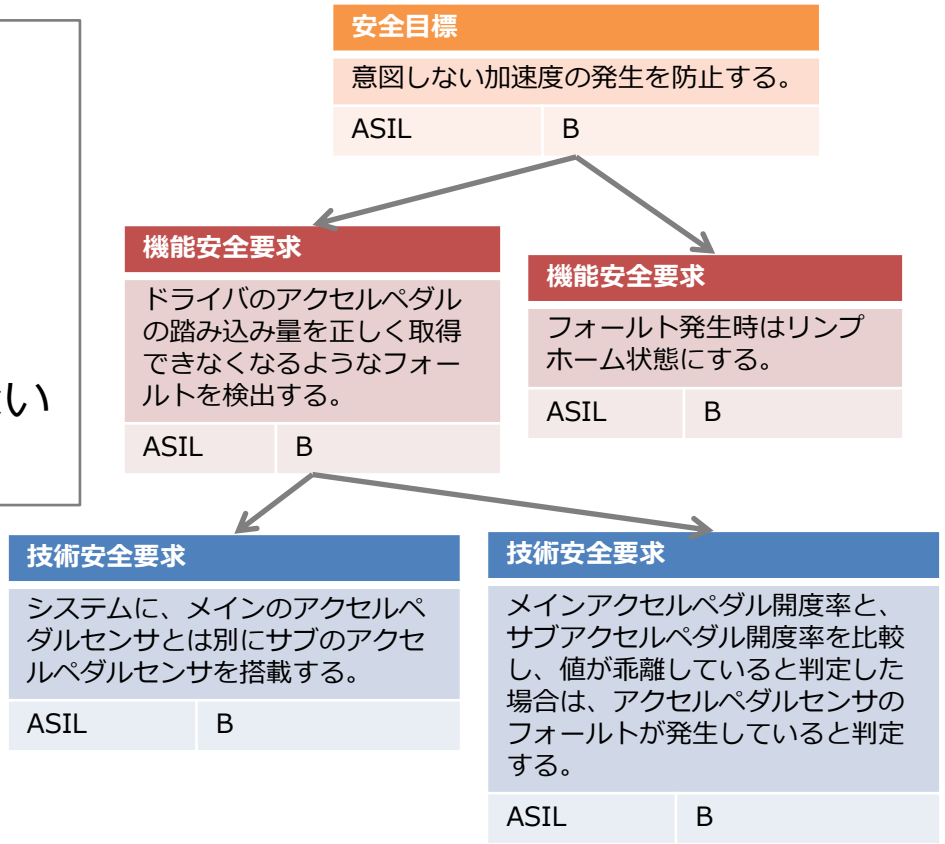
開発の早期に仕様の検証を行うことで、品質を向上できる

④自動車向け機能安全規格への対応がしやすくなる

- 自動車向けの安全規格であるISO26262では安全要求の定義、管理のしかたが規定されている

6.4.3.1 一組の安全要求は、
以下の特性を有していなければならない。
a)階層化構造
b)適切なグループ化方式に従う構造化
c)完全性
d)外部一貫性
e)階層構造の各レベル内に情報の重複がない
f)保守性

※ISO26262 Part8から抜粋



④自動車向け機能安全規格への対応がしやすくなる

- USDMの形式で安全要求を定義すると、規格の要件を満足できる

a)階層化構造を持つ　　b)「グループ」による構造化が可能

安全目標	SG.ENG.01	意図しない加速度の発生を防止する。			
	ASIL	B			
	機能安全要求	FSR.ENG.01	ドライバのアクセルペダルの踏み込み量を正しく取得できなくなるようなフォールトを検出する。		
		ASIL	C		
		技術安全要求	TSR.ENG.01	アクセルペダルセンサにセンサ値が異常な値になるようなフォールトが発生していることを検出する。	
			ASIL	C	
			理由	アクセルペダルセンサにフォールトが発生した時に、急加速を発生させないようにするため。	
			説明	センサにフォールトが発生すると、信号が入力されなかったり、実際の踏み込み量とは無関係の値が入	
		<H/Wへの要求：アクセルペダル踏み込み量の計測>			
		サブアクセルペダルセンサの搭載	技術安全要求	TSR.ENG.01.01	システムに、メインのアクセルペダルセンサとは別にサブのアクセルペダルセンサを搭載する。
				ASIL	C
		サブアクセルペダルセンサによるアクセ	技術安全要求	TSR.ENG.01.02	メインアクセルペダルセンサで、ドライバによるアクセルペダルの踏み込み量を計測する。
				ASIL	QM（B）
		サブアクセルペダルセンサによるアクセ	技術安全要求	TSR.ENG.01.03	サブアクセルペダルセンサで、ドライバによるアクセルペダルの踏み込み量を計測する。
				ASIL	B（B）

c)上位要求に対する完全性を確認しやすい

d)要求間の矛盾を見つけやすい

e) 情報の重複を見つけやすい

f)要求の追加や削除が容易に行える

⑤アーキテクチャ設計の質が向上する

- 制御仕様は「ソフトウェアの保守のしやすさ」を考慮して書かれてはいない
- そのため、制御仕様書の章立てのままソフトウェアを作ると、保守しにくいソフトウェアになる

01. ××制御

1. ××実施判定
   制御内容
   1. 以下が全て成立したら、処理1を行う
      ・sw == 1
      ・rev > START_REV
      ・count > START_TIME
   2. それ以外は処理2を行う

   処理
   1. startFlg = 1
   2. startFlg = 0

2. XX制御
   ・・・

02. ○○制御

・・・
3. ○○実施判定
   制御内容
   1. 以下が全て成立したら、処理1を行う
      ・sw == 1
      ・rev2 > START_REV
      ・count2 > START_TIME
   2. それ以外は処理2を行う

   処理
   1. startFlg2 = 1
   2. startFlg2 = 0
・・・

実は同じ処理をしている

この構成のまま
ソフトウェアを作ると、
始動判定方法が変わったら
複数個所修正が必要

保守性が悪化

⑤アーキテクチャ設計の質が向上する

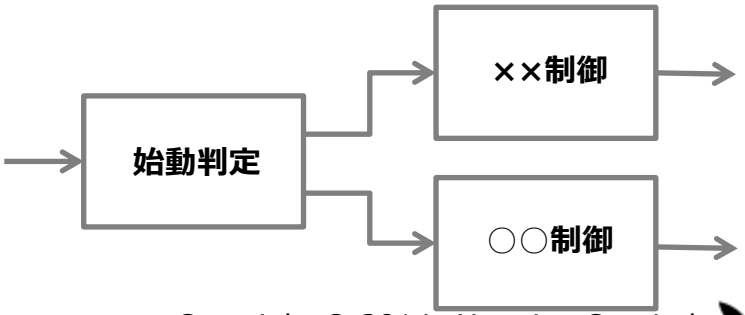
- 要求仕様に基づき、関係の強い機能が同じモジュールになるように設計することで、凝集度を高められる
- 凝集度を高くすることで保守しやすい（つまり派生開発しやすい）アーキテクチャにできる

サブシステム要求	XX.01.01	始動後はXX制御を行う	
	理由	・・・	
	説明	・・・	
	<始動判定>		
ソフト要求	XX.01.01.01	始動したことを判定する。	
	理由	・・・	
	説明	・・・	
	<始動判定>		
	SP.XX.001.01	以下の条件が全て成立した場合に、始動後であると判定する。 ・イグニッションスイッチが「オン」である ・エンジン回転数が所定値より大きい ・起動からの経過時間が所定値より大きい	
	SP.XX.001.02	上記の条件が成立していない場合は、始動前であると判定する。	
<XX制御>			
ソフト要求			

サブシステム要求	OO.01.01	始動後はOO制御を行う	
	理由	・・・	
	説明	・・・	
	<始動判定>		
ソフト要求	OO.01.01.01	始動したことを判定する。	
	理由	・・・	
	説明	・・・	
	<始動判定>		
	SP.OO.001.01	以下の条件が全て成立した場合に、始動後であると判定する。 ・イグニッションスイッチが「オン」である ・エンジン回転数が所定値より大きい ・起動からの経過時間が所定値より大きい	
	SP.OO.001.02	上記の条件が成立していない場合は、始動前であると判定する。	

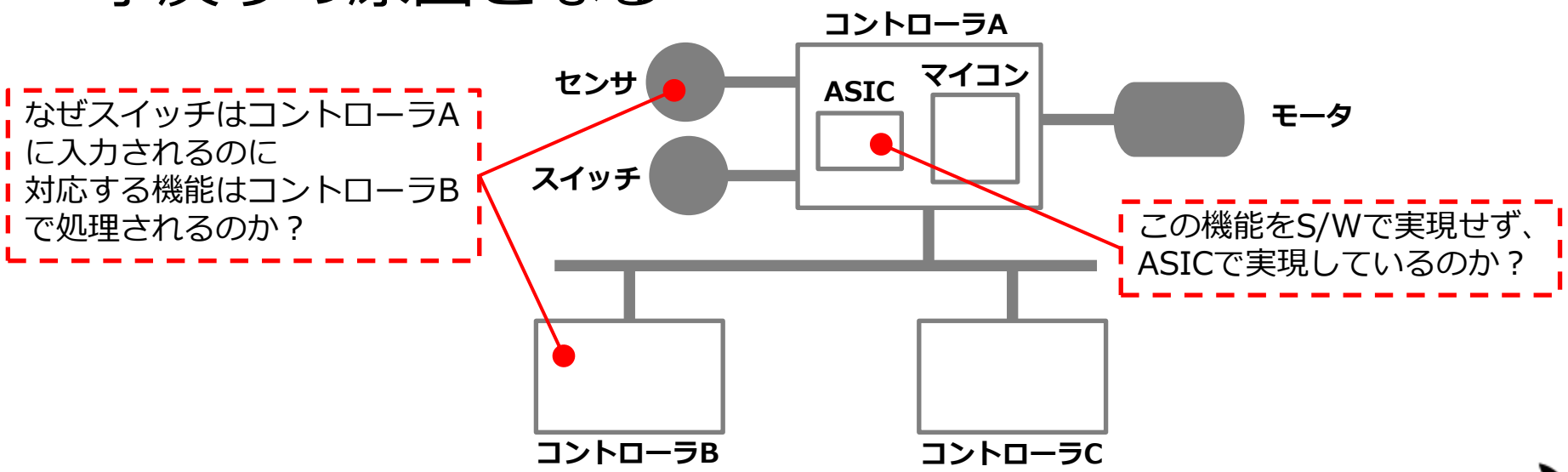
要求を明確にすることで、
同じ処理をしていることに気付ける

保守性を考慮して
ソフトウェア構造を検討できる



⑥システム設計の質が向上する

- システム設計は、H/W開発者、S/W開発者がすり合わせによって決め、成果物がない場合も多い
- しかしシステム設計も、なぜそのようなシステムになっているかという思想が明確でないと、派生開発をする中で思想に合わない変更をしてしまい、手戻りの原因となる



⑥システム設計の質が向上する

- システム設計でもシステム要求仕様を作成することで、システムの思想を表現することができる

システムレベルの要求仕様書

クルコンシステム			要求と要求仕様	対象サブシステム	
				クルコン	エンジン
システム要求	ACC.01		追従走行を行う場合、設定した車間設定に従いドライバがアクセルペダルを踏まなくても自動的に先行車との車間距離を保つように加速、減速を制御する。定速走行を行う場合は、設定された上限速度を保つように加速、減速を制御する。また、設定した車間設定、上限速度は変更できるようにする。		
	理由		運転時、ドライバがアクセルペダルを操作する負荷を無くし、楽に運転できるようにするため。		
	説明		国土交通省発行の『運転支援の考え方』で、運転支援システムに対して『強制介入できること』等のガイドラインが示されている。		
	<クルコンの始動>				
クルコンON	システム要求	ACC.01.01	ドライバがクルコンのONを要求したときに、クルコンをONできる条件が成立していたら、クルコンをONにしドライバに通知する。	○	
		理由	クルコンを使わない状況では、動作しないように機能をOFFできるようにする必要がある。		
		説明	特になし。		
			<クルコンON要求の判定>		
		□□□	SP.ACC.01.01	クルコンがOFFの時にドライバがON/OFFスイッチを…合、ドライバがクルコンのONを要求しているとする	

コンポーネントを
組み合わせたシステム
に関する要求仕様を書く

コンポーネントごとに、
そのコンポーネントに関する
要求仕様を書く

コンポーネントレベルの要求仕様書

クルコンコンポーネント			要求と仕様	
	<クルコンの始動>			
クルコンON	サブシステム要求	ACC.01.01	ドライバがクルコンのONを要求したときに、クルコンをONできる条件が成立していたら、クルコンをONにしドライバに通知する。	
		理由	クルコンを使わない状況では、動作しないように機能をOFFできるようにする必要が	
		説明	特になし。	
		<ON要求の判定>		
	ソフト要求	ACC.01.01.01	スイッチのチャタリングの影響を考慮し、ON/OFFスイッチが押下されたことを判定する。	
		理由	チャタリングによって誤判定しないようにするため。	
		説明	特になし。	
	□□□	<押下判定>		
		SP.ACC.001.01	ON/OFFスイッチの信号が「オン」である状態が50[ms]以上継続した後に「オフ」に変化した場合、ON/OFFスイッチ押下判定	

**システムレベルの要求仕様書に
記載された要求仕様を
コンポーネントレベルの要求仕様書で
詳細化する**

まとめ

まとめ

- 自動車メーカから部品メーカに入力される成果物が制御仕様のみが書かれた文書だけである
- そのため仕様の認識間違いによる不具合が増加
- ボトムアップのアプローチによって、制御仕様からUSDMの形式で要求を文書化した
- それにより不具合を削減できるような様々な効果が得られた
- 機能安全規格対応やシステム設計への応用、アーキテクチャ設計、テストへの入力のように、要求仕様を起点に上流から下流までをシームレスにつなぐことができることが分かった