手戻りのない開発を目指して

-USDMとの出会い-

株式会社ホンダロック 電装BL 有村 文宏

Agenda

1.会社紹介

2.従来のソフトウェア開発における要求仕様定義

3.改善に向けた取り組み

4.USDMを使用した際の課題

5.機能安全とUSDM

6.成果

7.今後の展開

1.会社紹介

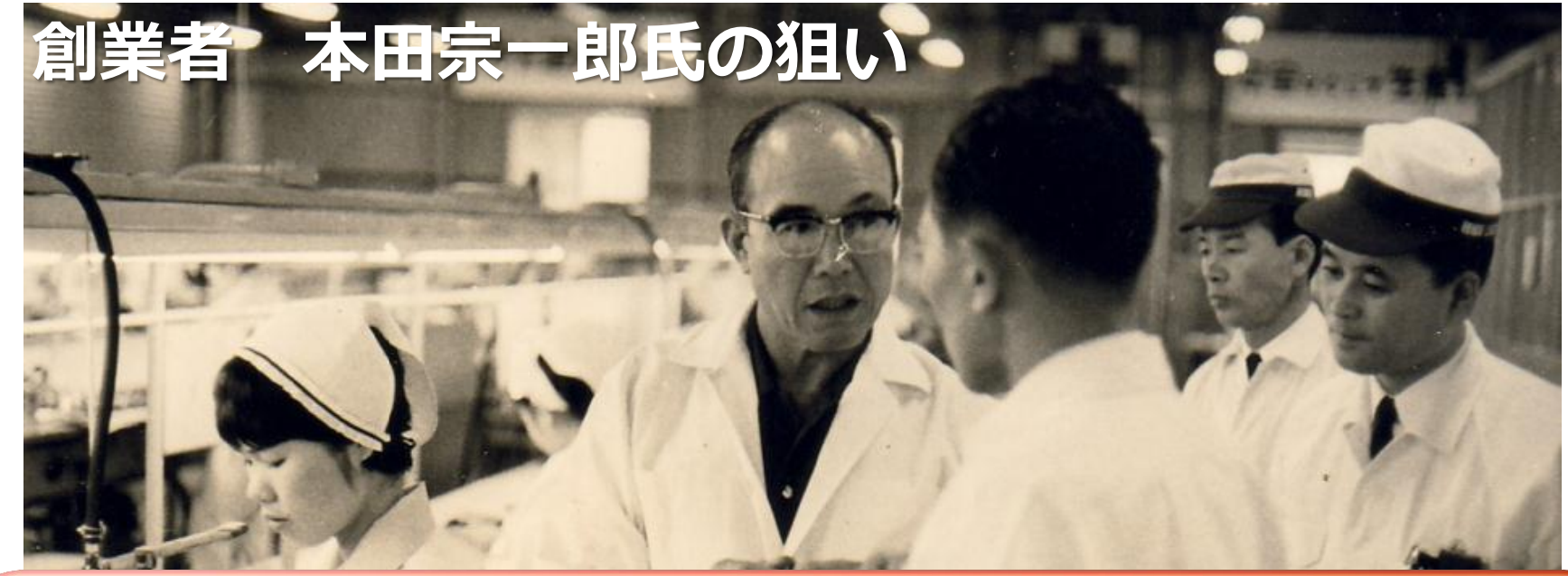
1.会社紹介

会社名　：株式会社ホンダロック（ホンダグループ）
設立　　：1962年4月5日
資本金　：21.5億円
本社　　：宮崎県宮崎市佐土原町
株主　　：本田技研工業 株式会社　１６４３千株（１００%）
拠点　　：国内5拠点、世界15拠点
国内開発拠点：栃木R&Dセンター（栃木県高根沢町情報の森内）、宮崎本社

栃木　R&Dセンター

国内5拠点、世界15拠点のグローバル企業です

創業者　本田宗一郎氏の狙い



地域振興の先導役を

　宮崎に近代工業が栄えないわけがない。我々は勇気を持って近代工業をこの地に起こす。
宮崎に見事な機械工業が花咲いて今後次々と立派な企業が進出してくることを希望する。

世界に通用するキーロックメーカー

競争相手は世界的メーカー。

距離が時間に置きかえ得ないはずはない。時間は知恵で解決できる。

困難な環境こそ最良のアイディアの条件である。宮崎に基盤を作り世界に打って出ること。

創業者本田宗一郎氏の思いを受け継ぎ現在に至っています

1.会社紹介


・アウターハンドル
・電動ステアリングロックなど


・ドアミラー
・リアビューミラーなど

セキュリティ・エントリー系

視界系

2輪製品

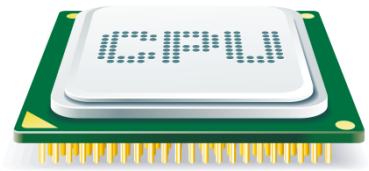
センサー系


・スマートエントリーシステム
・イモビライザーなど


・EPSトルクセンサー
・ABSホイルセンサーなど

4輪/2輪車のボディ電装系製品の開発・製造を行っています

ホンダロック製品のソフトウェア開発における特長



ソフトウェア開発規模
・使用CPU：8~16bit
・使用ROM容量：8~64kByte

①高セキュリティ性　例）イモビ・キーレスシステム

仮に誤動作した場合・・・
・車両が盗まれる！
・ドアロックの施錠・解錠をすることが出来ない！

②高信頼性　例）電動ステアリングロック

仮に誤動作した場合・・・
走行中にステアリングロックピンが飛び出て重大な事故に繋がる！

高セキュリティ性・高信頼性が求められる製品の
ソフトウェア開発を行っております

2.従来のソフトウェア開発における要求仕様定義

2.従来のソフトウェア開発における要求仕様定義

製品の担当者スキルに依存したソフトウェア開発を行っていた

2.従来のソフトウェア開発における要求仕様定義

①担当者でしか知らない要求仕様がある

- 第三者による検証で見落とされるため要求・仕様漏れが起きる
- 要求に対する仕様の妥当性がわからない

②記述粒度がおおざっぱ

- 第三者が理解するまでに時間を要する
- 担当者が変わった場合に引き継がれない情報がある

③記述方法が統一されていない

- レビュー時に見づらい
- 内容を把握するまでに時間が掛かる

従来の開発手法では要求仕様を定義する際に課題を抱えていた

3.改善に向けた取り組み

改善活動を行うきっかけは・・・・

重要保安部品の開発を行った際にOEM（ここでは自動車メーカー）から下記の説明を求められた

要求仕様が明確に定義されていないと説明出来ない

- 要求に対するソフトウェア仕様の妥当性証明
- ソフトウェア検証の網羅性

※重要保安部品とは故障した場合などに生命の危険に繋がるような部品のこと

従来のソフトウェア開発では対応することが困難

改善策として

①ソフトウェア開発プロセスの明確化

- 担当者によってプロセスを変えない
- ソフトウェア開発プロセスを用いることで各工程における作業を規定し、どういった成果物を残す必要が有るかを明確にする

②要求仕様をいかに正確に定義するか

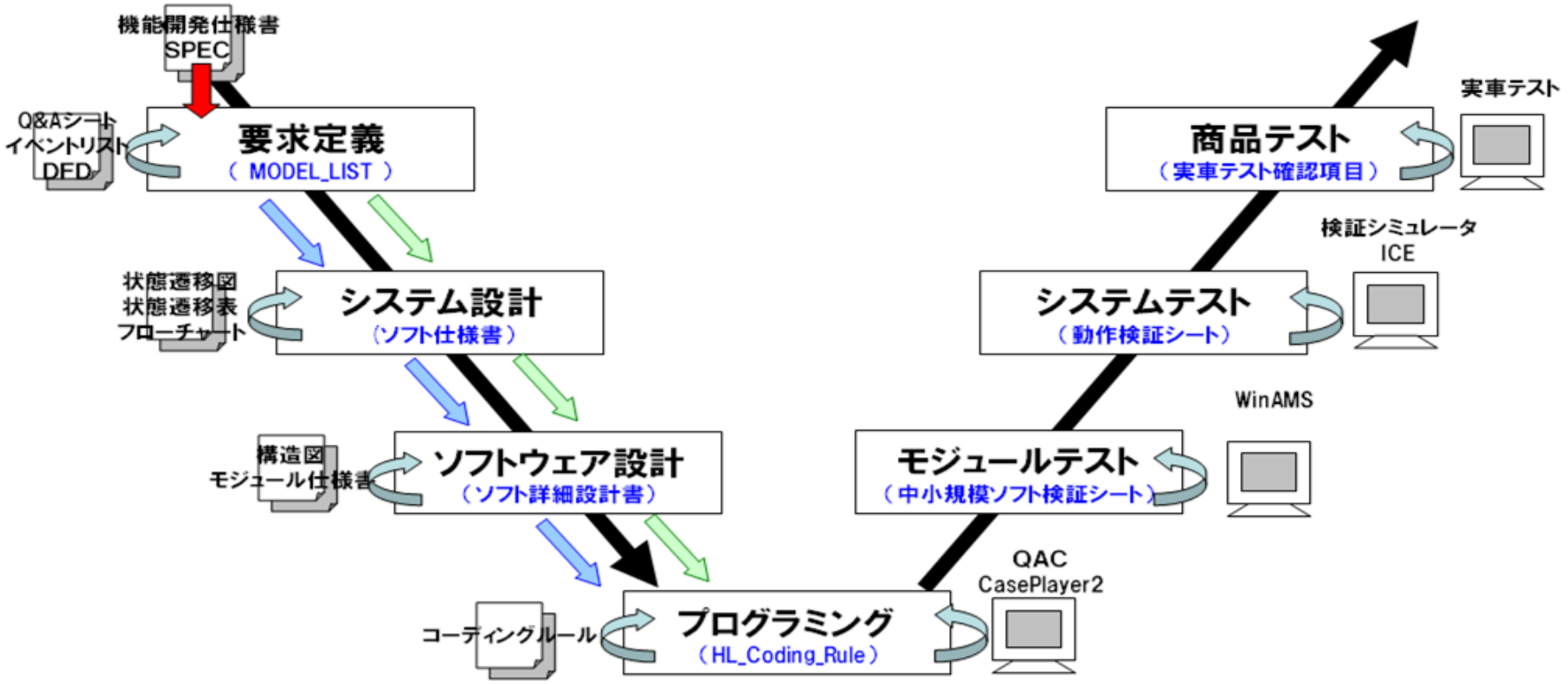
- タイミングチャートや図を使用して正確に要求仕様を定義する
- 要求仕様から導出されたソフトウェアとの一致性を明確にすること
- 手戻りを少なく出来る様な仕組みが必要

③要求仕様書の精度・見易さの向上

- 要求仕様書のフォーマットを統一する
- 自然言語のみではなく図を多用出来るようなフォーマットにする

改善に取り組むに当たり目標を設定

①ソフトウェア開発プロセスの明確化



※改善活動当初のもの

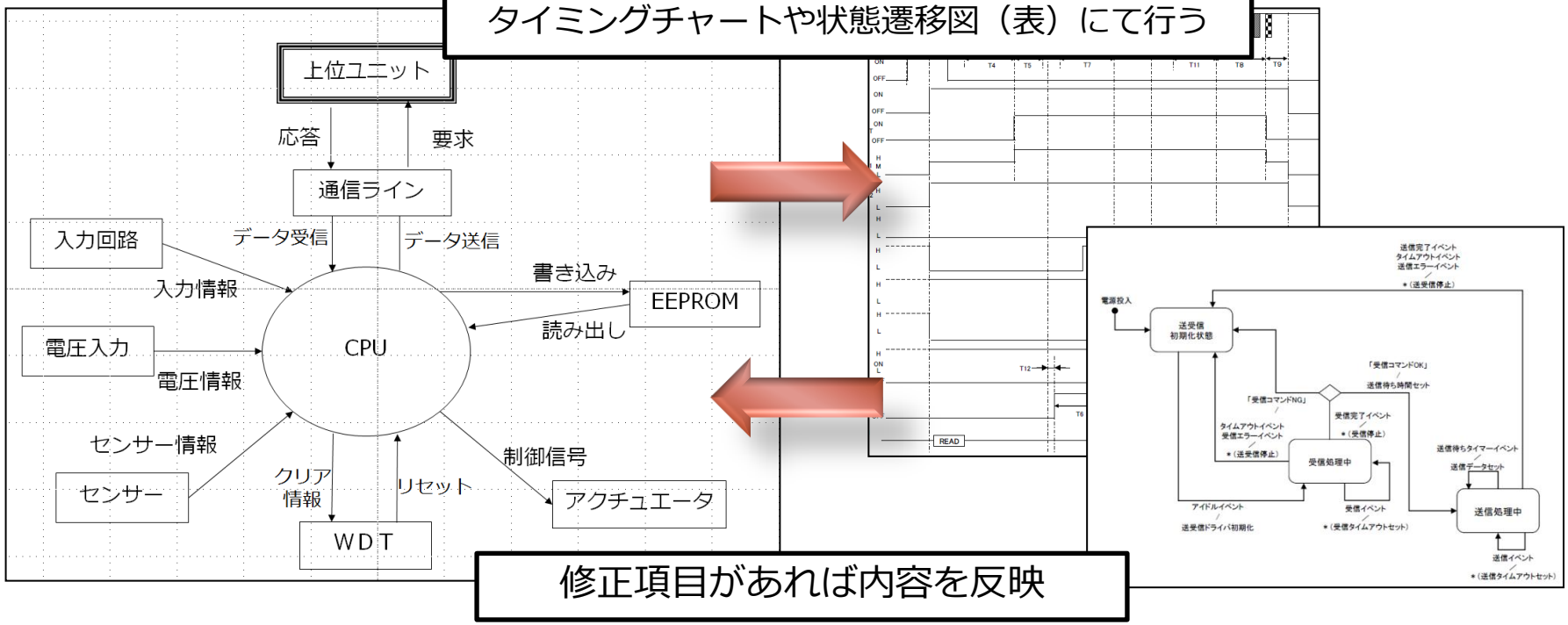
ソフトウェアプロセスの改善としてV字プロセスを導入

②上流工程をいかに正確に定義するか

客先からの要求

・内容に対してコンテキストダイアグラムを用いてシステムに対するイベントや入出力情報を明確にする（この段階で不足している項目をなくす）
・抽出したイベント情報を基にしてどのような振る舞いをするか確認する

イベント情報を基に振る舞いの確認を
タイミングチャートや状態遷移図（表）にて行う



修正項目があれば内容を反映

要求仕様定義の段階でシステムに対するイベントや入出力情報を明確にすることでシステムの振る舞いを早期に確認する

改善活動の成果

①ソフトウェア開発プロセスの明確化

- V字プロセスを適用することで各工程の役割、成果物の統一化を図った

②上流工程をいかに正確に定義するか

- DFDやイベントリスト、状態遷移図（表）を使用することでシステムの振る舞いを早期に検証

③要求仕様書の精度・見易さの向上

- ファイル形式と記述内容の統一を図り、可読性の向上に繋げた

改善活動後の新たな課題

②上流工程をいかに正確に定義するか

- 要求と仕様の繋がりを表現することが出来ていない

③要求仕様書の精度・見易さの向上

- 要求の妥当性が判断しにくい
- 連携が必要な外部ECUとシステム間の関係が表現出来ていない

改善活動を行った際に抽出された課題を分類すると

表現に対する課題

連携が必要な外部ECUとシステム間の関係が表現出来ていない

要求仕様に対する課題

・要求の妥当性が判断しにくい
・要求と仕様の繋がりを表現することが出来ていない

改善案①　UMLの活用

・統一化された表現手法で様々な側面を表現することが可能

改善案②　USDMの活用

・要求に対して仕様を階層的に表現することが出来る

・フォーマットの統一

抽出された課題に対してUMLとUSDMを活用して更なる改善を実施

3.改善に向けた取り組み

課題 連携が必要な外部ECUとシステム間の関係が表現出来ていない

対策 システムの機能を図を使用してわかりやすく表現することが出来るユースケース図を使用する

従来の記述方法

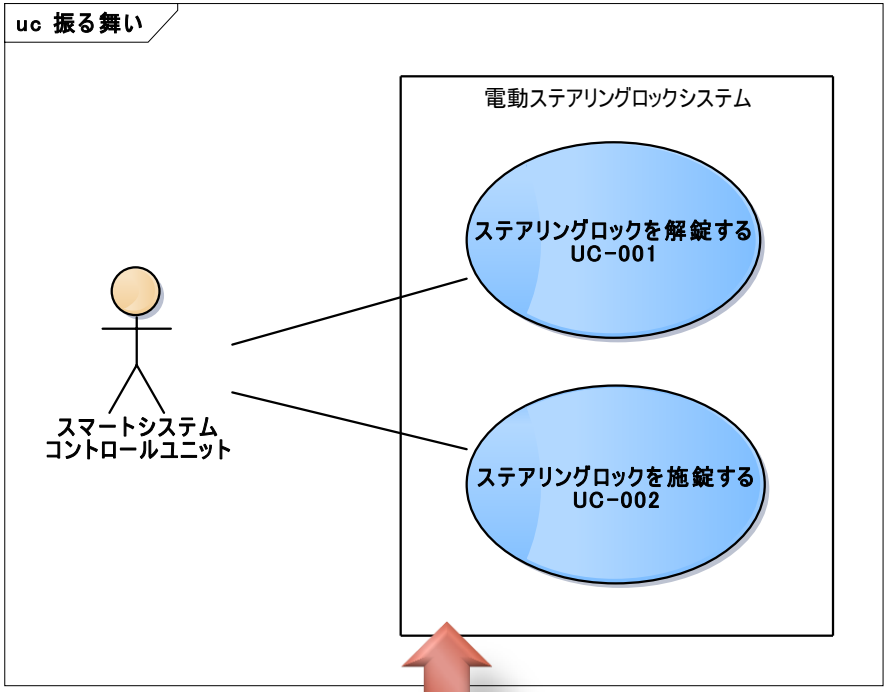
客先からの要求書に記述される機能を
そのまま要求仕様書へ記述

例）
①＊＊＊はステアリングロック施錠
動作を行うこと

②＊＊＊はステアリングロック解錠
動作を行うこと

自然言語でも理解することは可能であるが、システム全体としてどのECUとやり取りするのかが把握しづらい

ユースケース図を利用



図で明示することでシステム全体としての繋がりもわかりやすい

ユースケース図を使用することでシステムと外部ECUの繋がりを図で表現することが出来るため、理解しやすくなる

要求の抜け漏れの対策としてユースケース記述を作成する

ユースケース番号			UCD-01
ユースケース名			車両を＊＊＊＊＊状態にする
概要			ユーザーが＊＊＊＊動作を行うことで車両を＊＊＊状態にするため
主アクタ			ユーザー
副アクタ			なし
システム			ユニットA
前提条件			システムに電源が供給され、正常動作可能なこと
事後条件			＜正常＞ 車両が＊＊＊＊状態になる
フロー	条件	STEP	ステップ
基本フロー（B）		1	ユーザーは＊＊＊＊＊動作を行う。
		2	ユニットAは＊＊＊＊＊＊を＊＊＊する。※1 <A1><A2>
		3	ユニットAは＊＊＊＊＊となった場合、＊＊＊＊を規定時間行う。※2<A3>
		4	ユニットAは規定時間経過後、＊＊＊＊をOFFにする。＜E1＞
		5	ユニットAは間欠動作を行う。＜E2＞
		6	B1に戻る。
代替フロー（A）	A1:＊＊＊＊の場合	1	ユニットAは間欠動作を行う。※3
		2	<B2>に戻る。
	A2:＊＊＊＊の場合	1	ユニットAは間欠動作を行わない。※4
		2	<B3>に戻る。
	A3:＊＊＊＊の場合	1	ユニットAは規定時間変化量を規定時間監視する。＜A4>※5
		2	<B3>に戻る。
例外フロー（E）	E1:＊＊＊＊の場合	1	ユニットAは＊＊＊＊を実行し、＊＊＊＊させる。※3
	E2:＊＊＊＊の場合	1	ユニットAは＊＊＊＊を停止し、＊＊＊＊させる。※3
備考(*)		1	＊＊＊＊＊＊
		2	＊＊＊＊＊＊
		3	＊＊＊＊＊＊
		4	＊＊＊＊＊＊
		5	＊＊＊＊＊＊

基本的なシステムの振る舞いを定義

条件分岐があった場合の振る舞いを記述する

エラー時における処理を記述する

注記する内容があれば記述する

ユースケース記述を作成し、早期にシステムとしての振る舞いを検証することが可能であり要求の抜け漏れを防ぐことが出来る

3.改善に向けた取り組み

課題	要求の妥当性が判断しにくい。また、要求と仕様の繋がりを表現することが出来ていない。
対策	要求に対する理由を明確に記述出来、要求と仕様の繋がりを持たせやすいUSDMを使用する

従来の要求仕様の記述方法

要求：システムの電源投入後、＊＊＊ms後に動作可能であること
仕様：リセット解除信号から通信可能となるまで＊＊＊ms未満とし、＊＊＊ms以内に内部の情報を取得し、上位ユニットから送信されるデータに対して返信出来る状態にする。

従来の記述方法ではなぜこのような要求が有るのかといったことが不明確であった。また、複数の仕様が合わさって表現されるような場合はわかりにくい記述となってしまうため、USDMを使用することで改善を行う

USDMの使用　①要求と仕様を階層的に表現する

利点

・要求に対する仕様を把握しやすい
・要求に対する理由が明示されることで要求や仕様に対する理解をしやすい
・グループ化することで関連する仕様を把握しやすい
（複数の仕様をわかりやすく表現可能）

要求と要求仕様		
要求	ID.01	車両を＊＊＊状態にする。
	理由	ユーザーが＊＊＊＊＊を実施した場合、＊＊＊＊＊動作を行うため。
	説明	車両を＊＊＊状態にし、ユーザーに＊＊＊機能を提供するため。
＜機能要求＞		
要求（機能）	ID.01.01	ユニットAは＊＊＊＊＊を＊＊＊すること。
	理由	ユーザーからの＊＊＊動作を検出するため。
	説明	ユーザーが＊＊＊動作を実行した場合、車両を＊＊＊状態にして＊＊＊機能をユーザーに提供するため。
＜＊＊＊判定＞		
□□□	ID.01.01.001	ユニットAは＊＊＊入力を＊＊＊ms周期で判定する。
＜＊＊＊入力がHiであった場合＞		
□□□	ID.01.01.002	ユニットAは＊＊＊と判断し、入力確定処理を行う。
＜＊＊＊入力がHi確定した場合＞		
□□□	ID.01.01.003	ユニットAは＊＊＊信号を＊＊＊ms後にCANで送信する。
＜＊＊＊入力がLoであった場合＞		
□□□	ID.01.01.004	ユニットAは＊＊＊動作を継続する。

②要求に対する理由を記述することが出来る

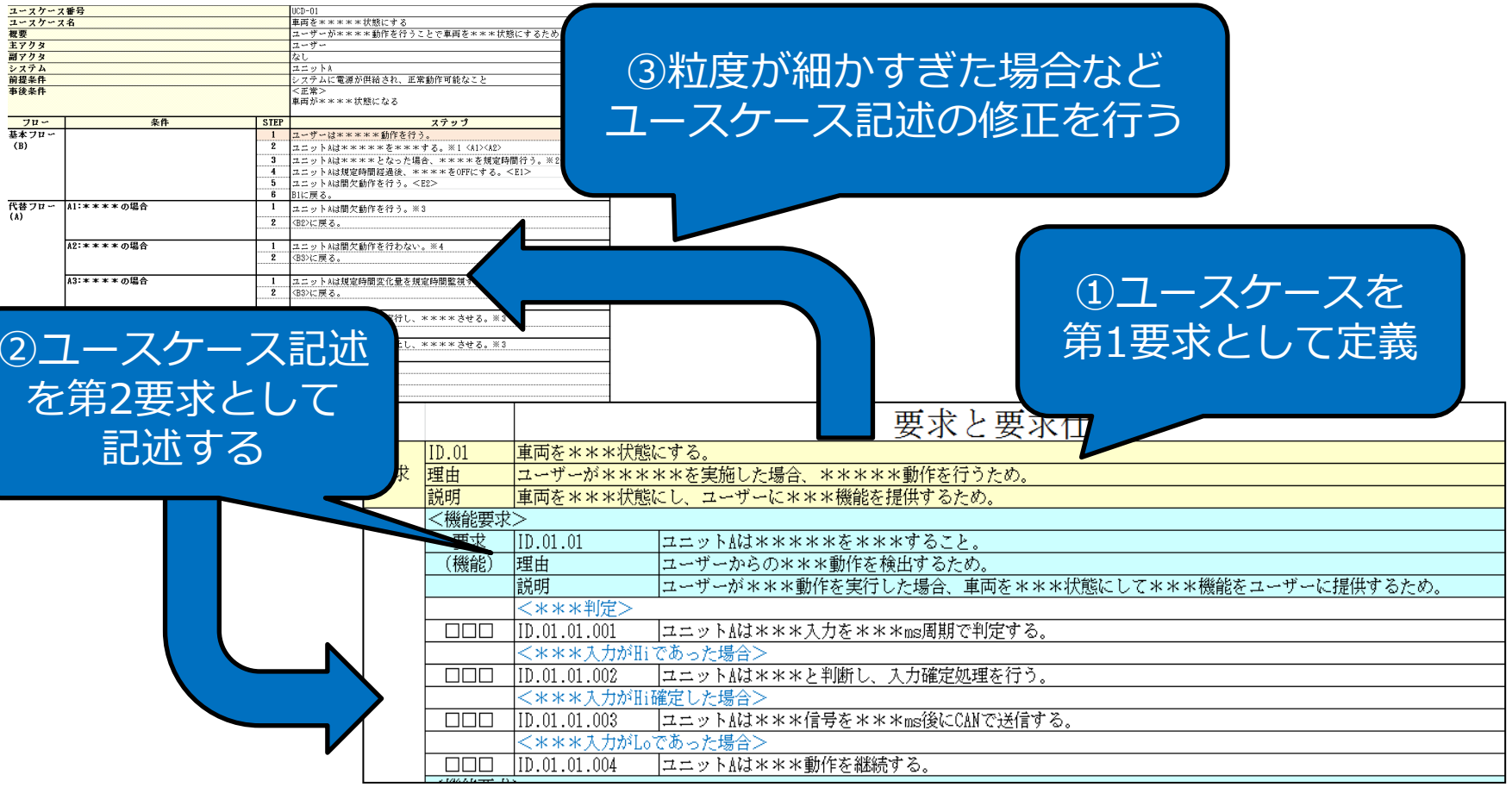
③関連する仕様をグループ化する

①階層的に表現することが出来るため、要求に対する仕様を把握しやすい

USDMの使用　②システムの振る舞いと要求仕様の繋がり

利点

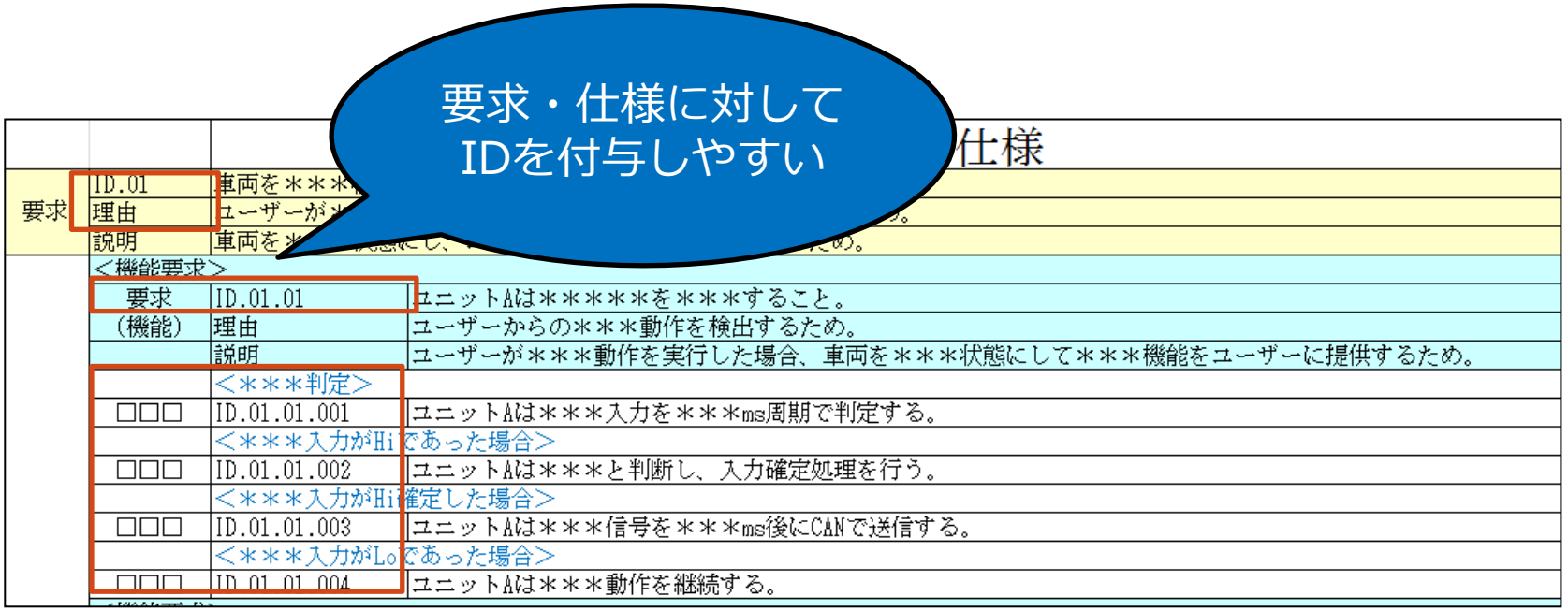
・システムの振る舞いと要求仕様に繋がりを持たせることが出来る
・システムの振る舞いに対する要求と仕様を理解しやすくなることで抜け漏れを抽出しやすい



USDMの使用　③トレーサビリティを確保しやすい

利点

・要求や仕様にIDを付与することが出来るためトレーサビリティを確保しやすい



			仕様
要求	ID.01	車両を＊＊＊	
	理由	ユーザーが＊	。
	説明	車両を＊	ため。
	＜機能要求＞		
	要求	ID.01.01	ユニットAは＊＊＊＊＊を＊＊＊すること。
	（機能）	理由	ユーザーからの＊＊＊動作を検出するため。
		説明	ユーザーが＊＊＊動作を実行した場合、車両を＊＊＊状態にして＊＊＊機能をユーザーに提供するため。
		＜＊＊＊判定＞	
	□□□	ID.01.01.001	ユニットAは＊＊＊入力を＊＊＊ms周期で判定する。
		＜＊＊＊入力がHiであった場合＞	
	□□□	ID.01.01.002	ユニットAは＊＊＊と判断し、入力確定処理を行う。
		＜＊＊＊入力がHi確定した場合＞	
	□□□	ID.01.01.003	ユニットAは＊＊＊信号を＊＊＊ms後にCANで送信する。
		＜＊＊＊入力がLoであった場合＞	
	□□□	ID.01.01.004	ユニットAは＊＊＊動作を継続する。

USDMの使用　④記述方法の統一化

利点

・形式的に記述することが出来るためテンプレートを作りやすい
・記述方法のルール化を図ることが出来るため記述内容を統一しやすい

①形式的に記述することが出来、テンプレートに適している

		要求と要求	
要求	ID.01	車両を＊＊＊状態にする。	
	理由	ユーザーが＊＊＊＊＊を実施した場合、＊＊＊＊＊動作を行うため。	
	説明	車両を＊＊＊状態にし、ユーザーに＊＊＊機能を提供するため。	
	＜機能要求＞		
	要求（機能）	ID.01.01	ユニットAは＊＊＊＊＊を＊＊＊すること。
		理由	ユーザーからの＊＊＊動作を検出するため。
		説明	ユーザーが＊＊＊動作を実行した場合、車両を＊＊＊状態にして＊＊＊機能をユーザーに提供するため。
		＜＊＊＊判定＞	
	□□□	ID.01.01.001	ユニットAは＊＊＊入力を＊＊＊ms周期で判定する。
		＜＊＊＊入力がHiであった場合＞	
	□□□	ID.01.01.002	ユニットAは＊＊＊と判断し、入力確定処理を行う。
		＜＊＊＊入力がHi確定した場合＞	
	□□□	ID.01.01.003	ユニットAは＊＊＊信号を＊＊＊ms後にCANで送信する。
		＜＊＊＊入力がLoであった場合＞	
	□□□	ID.01.01.004	ユニットAは＊＊＊動作を継続する。

②テンプレートを作りやすいためルール化しやすい

4.USDMを使用した際の課題

USDMを使用した際の課題　①記述の粒度

課題

・大ざっぱや細かすぎるといった点で意見が分かれる
・明確なガイドラインがないため、同じような記述があっても担当者により記述方法が異なる

USDMを使用した際の課題　②図との関係性

課題

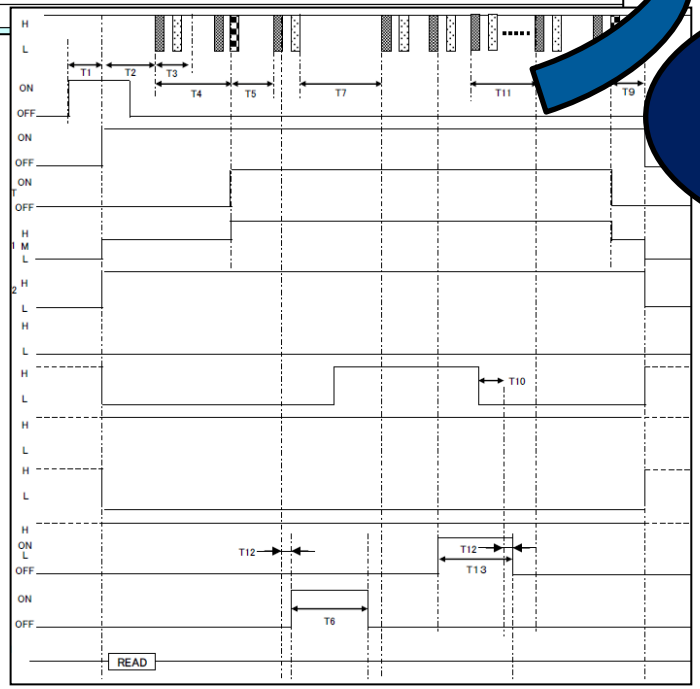
・エクセルで階層的に要求仕様を記述するため、別シートに記述している参照図を確認しづらい

		要求と要求仕様	
要求	ID.01	車両を＊＊＊状態にする。	
	理由	ユーザーが＊＊＊＊＊＊を実施した場合、＊＊＊＊＊動作を行うため。	
	説明	車両を＊＊＊状態にし、ユーザーに＊＊＊機能を提供するため。	
	＜機能要求＞		
	要求（機能）	ID.01.01	ユニットAは＊＊＊＊＊＊を＊＊＊すること。
	理由	ユーザーからの＊＊＊動作を検出するため。	
	説明	ユーザーが＊＊＊動作を実行した場合、車両を＊＊＊状態にして＊＊＊機能をユーザーに提供するため。	
	＜＊＊＊判定＞		
	□□□	ID.01.01.001	ユニットAは＊＊＊入力を＊＊＊ms周期で判定する。
	＜＊＊＊入力がHiであった場合＞		
	□□□	ID.01.01.002	ユニットAは＊＊＊と判断し、入力確定処理を行う。
	＜＊＊＊入力がHi確定した場合＞		
	□□□	ID.01.01.003	ユニットAは＊＊＊信号を＊＊＊ms後にCANで送信する。
	＜＊＊＊入力がLoであった場合＞		
	□□□	ID.01.01.004	ユニットAは＊＊＊動作を継続する。

②確認後、該当項目を探しながら元の位置に戻る

内容確認しながらシートも探したりして手間がかかるな・・・

①参照図のシートを確認



USDMを使用した際の課題　③要求仕様と下流工程のトレーサビリティ

課題　USDMと下流工程のトレーサビリティをどのように確保するか

		要求と要求仕様	
要求	ID.01	車両を＊＊＊状態にする。	
	理由	ユーザーが＊＊＊＊＊＊を実施した場合、＊＊＊＊＊動作を行うため。	
	説明	車両を＊＊＊状態にし、ユーザーに＊＊＊機能を提供するため。	
	＜機能要求＞		
	要求	ID.01.01	ユニットAは＊＊＊＊＊＊を＊＊＊すること。
	（機能）	理由	ユーザーからの＊＊＊動作を検出するため。
		説明	ユーザーが＊＊＊動作を実行した場合、車両を＊＊＊状態にして＊＊＊機能をユーザーに提供す
		＜＊＊＊判定＞	
	□□□	ID.01.01.001	ユニットAは＊＊＊入力を＊＊＊ms周期で判定する。
		＜＊＊＊入力がHiであった場合＞	
	□□□	ID.01.01.002	ユニットAは＊＊＊と判断し、入力確定処理を行う。
		＜＊＊＊入力がHi確定した場合＞	
	□□□	ID.01.01.003	ユニットAは＊＊＊信号を＊＊＊ms後にCANで送信する。
		＜＊＊＊入力がLoであった場合＞	
	□□□	ID.01.01.004	ユニットAは＊＊＊動作を継続する。

②確認後、該当項目を探しながら元の位置に戻る

制御モデル

①USDMを確認しながら制御モデルの内容を確認

制御モデルと要求仕様の繋がりがうまく表現出来ていないなぁ・・・

USDMを使用した際の課題　④作成工数について

課題

・トレーサビリティのためにIDを記述しているが、要求仕様を修正した場合の保守が大変
・見易さ向上のために設定したフォントの色などを忘れがち

要求	ID.01	車両を＊＊＊状態にする。	
	理由	ユーザーが＊＊＊＊＊を実施した場合、＊＊＊＊	
	説明	車両を＊＊＊状態にし、ユーザーに＊＊＊機能を提供す	
	＜機能要求＞		
	要求	ID.01.01	ユニットAは＊＊＊＊＊を＊＊＊すること。
	（機能）	理由	ユーザーからの＊＊＊動作を検出するため。
		説明	ユーザーが＊＊＊動作を実行した場合、車両を＊＊＊状 て＊＊＊機能をユーザーに提供するため。
		＜＊＊＊判定＞	
	□□□	ID.01.01.001	ユニットAは＊＊＊入力を＊＊＊ms周期で判定する。
		＜＊＊＊入力がHiであった場合＞	
	□□□	ID.01.01.002	ユニットAは＊＊＊と判断し、入力確定処理を行う。
		＜＊＊＊入力がHi確定した場合＞	
	□□□	ID.01.01.003	ユニットA＊＊＊信号を＊＊＊ms後にCANで送信する。
		＜＊＊＊入力がLoであった場合	
	□□□	ID.01.01.004	ユニットA＊＊＊動作を継続する。

仕様の追加！！

仕様の追加があった場合
追加以降の仕様IDを更新する必要がある！！

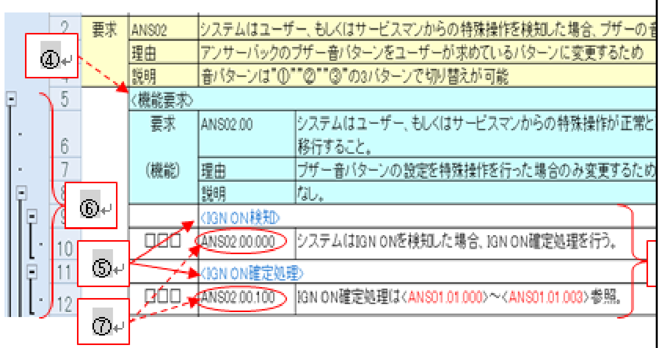
IDの更新が大変だなぁ・・・

4.USDMを使用した際の課題

課題	①記述の粒度
対策	記述の粒度や書き方、考え方を統一するためにガイドラインの作成・展開を実施

内容

2.5.→ 全体図



①→第1階層の要求。個々のユースケースに対応している。
②→第2階層の要求。ユースケース記述のステップに対応している。
③→仕様欄。ここに各要求を実現するための仕様を記載していく。
④→要求のグループ名。要求を括る場合に記載する。
⑤→仕様のグループ名。仕様の括りを意味する。
⑥→Excelのグループ化。要求、仕様、仕様のグループ名の表示方法を設定

3→ 作成のポイント

3.1.→ 考え方の軸(作成の判断基準)

- 要求を満たした仕様となっているか。
- 実装(Cコード)又はsimulinkモデルが想像できるレベルの記述となっているか。

3.2.→ 要求記述のコツ

- 動詞で表現したか→動詞で表現することで、その要求が表す範囲が明確になり仕様が導きやすくなる。
- 理由欄には要求の裏付けとなるような内容を記載したか→要求の勘違いを防ぎ、要求の目的をより明確にすることができる。
- 説明欄について、ソフトに直接関連がなくても必要に応じて説明を記載したか→理由欄と同様。

3.3.→ 仕様記述のコツ

- 仕様で書いた内容は要求で書かれた範囲内か。
- 仕様にあって要求にない仕様はないか。
- 要求の内容が仕様に表れているか。
- ある仕様のグループに属する仕様数は適切か(7±2となっているか)。
- 記載レベルが要求レベルではなく、仕様レベルとなっているか。
- 仕様のグループ化を行った際、その基準は妥当か。

3.4.→ 仕様抽出がはかどらない時の検討事項

- 要求で求めている範囲が見えるか。
- 要求の範囲が広すぎないか。

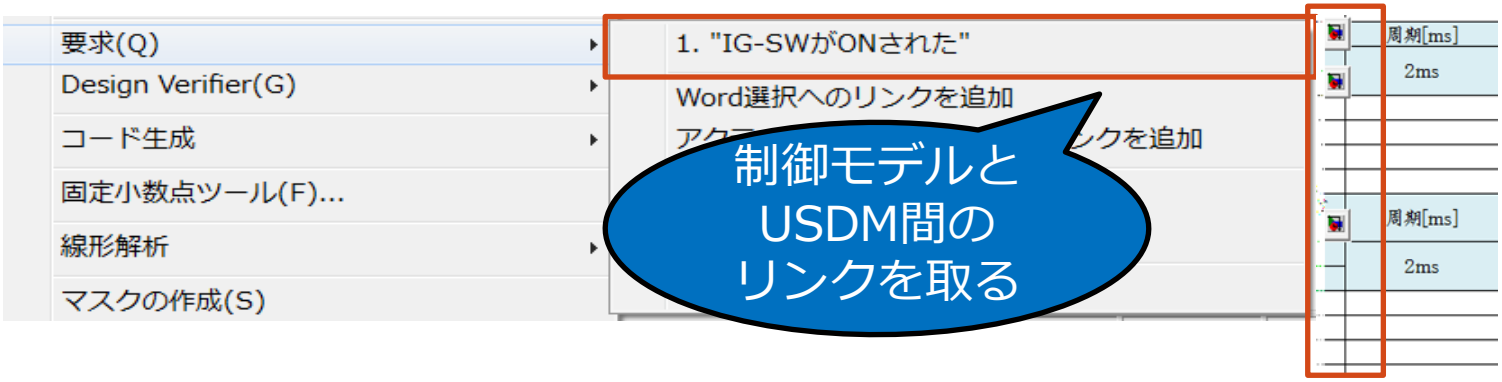
4.USDMを使用した際の課題

課題 ②図との関係

対策 リンクを活用することでユースケース記述や参照図を容易に確認出来る様に改善を実施

課題 ③要求仕様と下流工程のトレーサビリティ

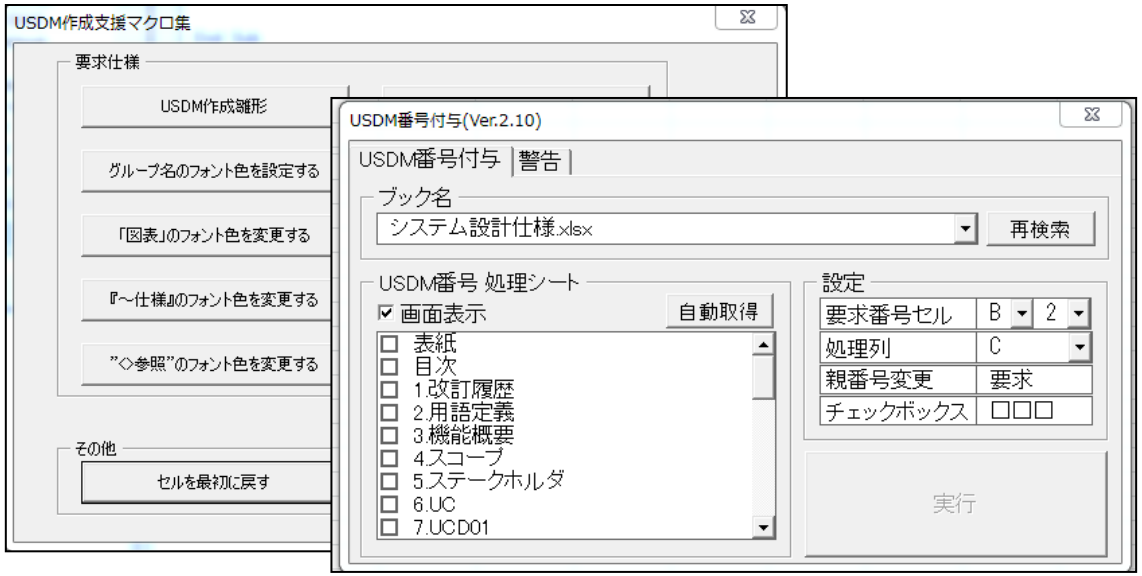
対策 USDMの仕様と制御モデルを市販のツールを使用してリンクさせ、要求仕様と制御ロジックとのトレーサビリティを確保する



4.USDMを使用した際の課題

課題 ④作成工数について

対策 単純作業はマクロ化することで自動化を図り、人的ミスを防ぐだけでなく作業工数の低減にも繋げる



自動生成を行うことで作業を簡略化する

		要求と要求仕	
要求	ID.01	車両を＊＊＊状態にする。	
	理由	ユーザーが＊＊＊＊＊を実施した場合、＊＊＊＊＊動作を行うため。	
	説明	車両を＊＊＊状態にし、ユーザーに＊＊＊機能を提供するため。	
	＜機能要求＞		
	要求	ID.01.01	ユニットAは＊＊＊＊＊を＊＊＊すること。
	（機能）	理由	ユーザーからの＊＊＊動作を検出するため。
		説明	ユーザーが＊＊＊動作を実行した場合、車両を＊＊＊状態にして＊＊＊機能をユーザーに提供するため。
		＜＊＊＊判定＞	
	□□□	ID.01.01.001	ユニットAは＊＊＊入力を＊＊＊ms周期で判定する。
		＜＊＊＊入力がHiであった場合＞	
	□□□	ID.01.01.002	ユニットAは＊＊＊と判断し、入力確定処理を行う。
		＜＊＊＊入力がHi確定した場合＞	
	□□□	ID.01.01.003	ユニットAは＊＊＊信号を＊＊＊ms後にCANで送信する。
		＜＊＊＊入力がLoであった場合＞	
	□□□	ID.01.01.004	ユニットAは＊＊＊動作を継続する。

5.機能安全とUSDM

自動車向け機能安全規格であるISO26262への対応

安全の種類

本質安全：根源からリスクをなくして達成される安全

機能安全：付与された機能によって確保される安全

例）
マイコンの誤動作による安全目標の侵害
本質安全：マイコンをなくす

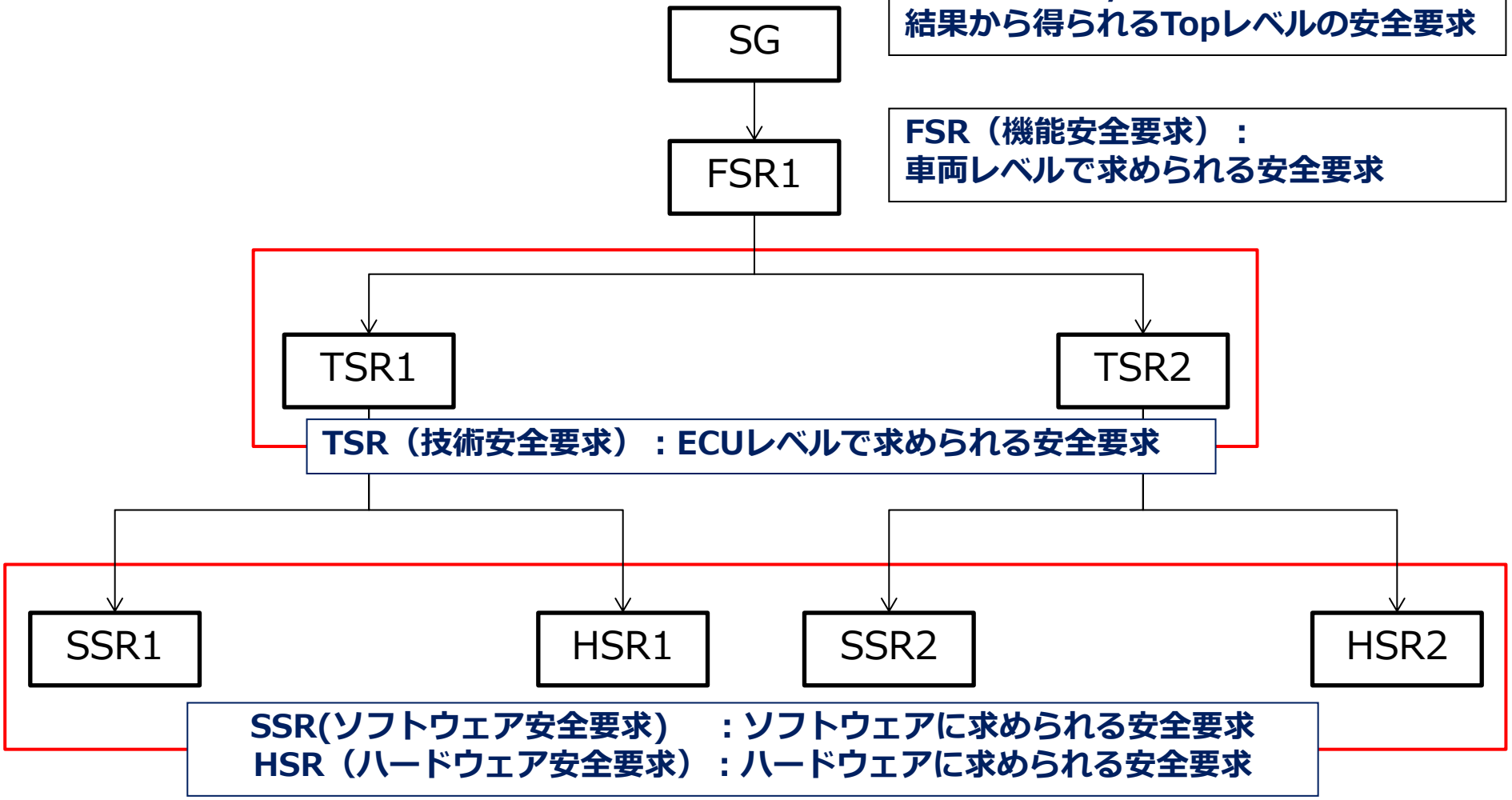
機能安全：マイコン監視機構、二重化など

ISO26262（以下機能安全）では危険に至る事象に対して何等かの手法を用いて安全な状態にするための要求を明確にする必要がある

機能安全における要求レベル

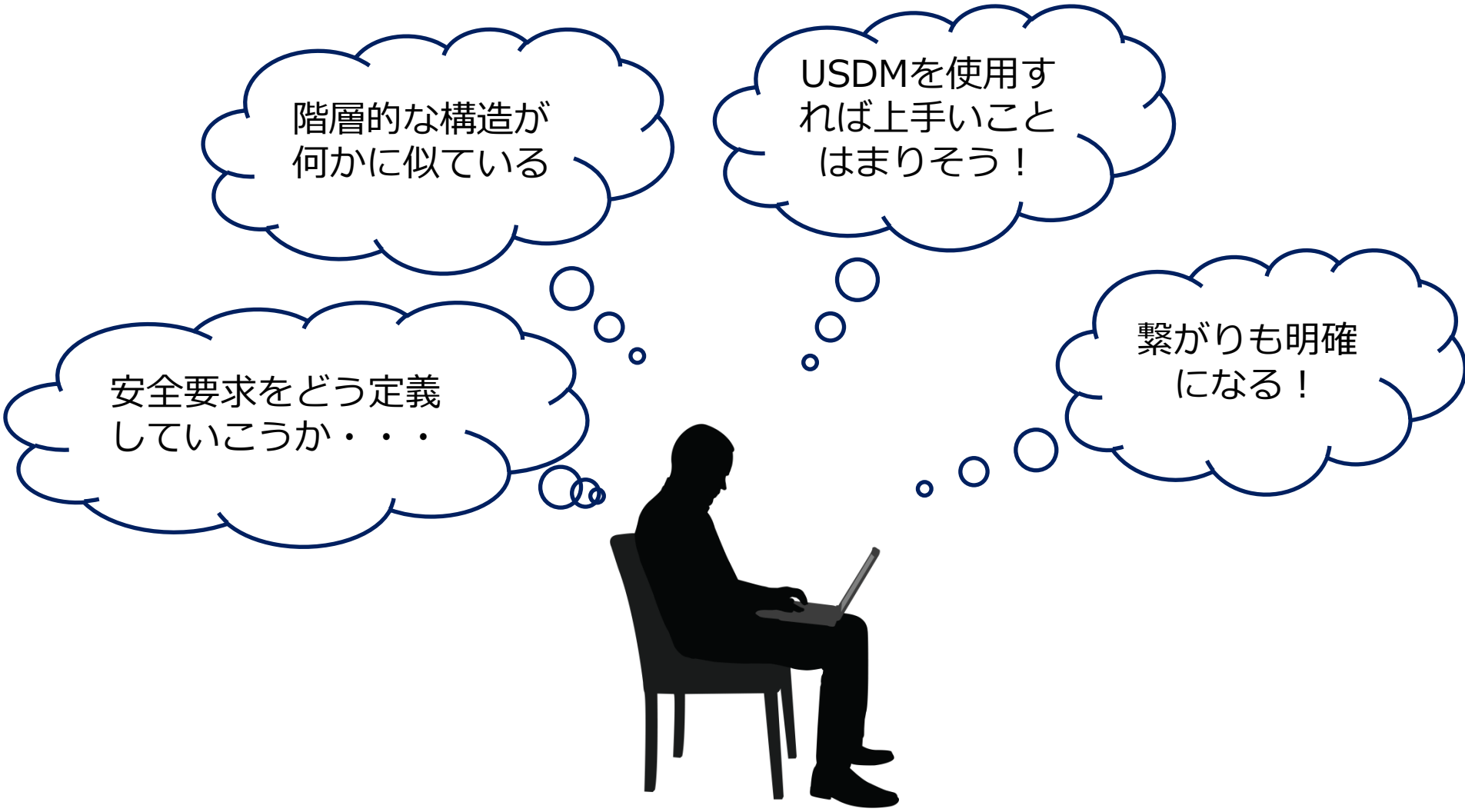
SG(セーフティゴール)：
ハザード分析 / リスクアセスメントの結果から得られるTopレベルの安全要求

FSR（機能安全要求）：
車両レベルで求められる安全要求



各要求レベルに応じた安全要求を記述する必要がある

機能安全にUSDMを使用するきっかけ



安全要求が階層的な構造になっていることに着目し、USDMを活用した

各安全要求の繋がりを表現する

	安全目標（SG）	機能安全要求（FSR）	技術安全要求（TSR）		ASIL
SG	SG 1b	＊＊＊の場合、＊＊＊をしてはいけない			B
	理由	異常動作防止のため			
	説明	本SGの安全状態は以下の通り。 ・＊＊＊をOFFとして制御する ・システムが動作できない故障が発生した場合、すべての出力を停止する。			
	FSR	FSR 1b-1	＊＊＊＊となった場合、＊＊＊をFTTI以内に検知すること		B
		理由	誤ってユーザーの＊＊＊＊動作と逆の動作をさせないようにするため。		
		説明	・＊＊＊＊判定に関しては＊＊＊の値と＊＊＊の値を使用すること。		
			＜異常判定＞		
	TSR	□□□	TSR 1b-1-1	ユニットB出力信号値の異常を判定すること。 【補足】SW確定データがデータ化けにより意図しないデータとなっていないかを確認する	B
	TSR	□□□	TSR 1b-1-2	ユニットCがCAN出力するデータの異常を判定すること。 【補足】CAN送信用データがデータ化けにより意図しないデータとなっていないかを確認する	
			＜演算器不具合検知＞		
	TSR	□□□	TSR 1b-1-3	WDTを配置すること。	B
	TSR	□□□	TSR 1b-1-4	ソフトウェアの実行の異常を検知すること 【補足】”ソフトウェアの実行の異常”とは、ソフトの暴走により、RAM化け、処理順異常が発生した状態を想定 【補足】WDT設定時間はソフトウェア詳細設計仕様を参照すること	B

第1要求にSGを記述

第2要求にFSRを記述

仕様にTSRを記述

SGからTSRまでの安全要求をUSDMとして定義し、繋がりを明確にする

TSRからSSR仕様までの繋がり

A	B	C	D	E
	技術安全要求 (TSR)	ソフトウェア安全要件 (SSR)	ソフトウェア安全仕様 (Software Safty Specification)	ASIL
TSR	TSR 3-1-2	２つのセンサの相関から故障検出を行うこと		B
	理由	単独では故障判定ができないため Part 5 Annex D Table D.11-センサー「センサーコリレーション」が典型的カバレッジ「高（99%）」であるため選択		
	説明	一定の範囲内であれば正常とする		
	SSR	SSR 3-1-2-1	相関が範囲外の場合、センサー故障と判定すること	B
		理由	単独の異常を検知したいため	
		説明	センサ電源電圧が正常時のみ判定する	
		＜初回相関判定＞		
	□□□	SSS 3-1-2-1-01	IG1 ON後に初回センサ確定値および初回ポジションセンサ電源電圧が取得できるまでは更新する情報はすべて初期値とする。	
		＜通常動作＞		
	□□□	SSS 3-1-2-1-02	センサ1とセンサ2の確定値を同一周期で取得する。	
	□□□	SSS 3-1-2-1-03	２つのセンサ確定値を加算する。	
	□□□	SSS 3-1-2-1-04	加算結果は次回加算タイミングまで保持する。	
		＜相関故障判定＞		
	□□□	SSS 3-1-2-1-05	加算結果が以下を満たすときに正常、そうでないときに相関故障と判定し、センサ故障の相関故障確定をセットする。 ※詳細は顧客から提示された仕様書に従うこと	

第1要求にTSRを記述

第2要求にSSRを記述

仕様にSSR仕様を記述

USDMを使用することでSGからSSR仕様までの
繋がりを保ちながら定義していくことが可能となる

機能安全へ適用する際の工夫点

A	B	C	D	E	F	G	H	I	J
	安全目標（SG）	機能安全要求（FSR）	技術安全要求（TSR）		ASIL	安全方策	システムレベル	独立配置要求	システム設計仕様（該当シナリオまたは該当STEP）
SG	SG 1b	＊＊＊の場合、＊＊＊をしてはいけない			B				SYS-D-03
	理由	異常動作防止のため							
	説明	本SGの安全状態は以下の通り。 ・＊＊＊をOFFとして制御する ・システムが動作できない故障が発生した場合、すべての出力を停止する。							
	FSR	FSR 1b-1	＊＊＊＊となった場合、＊＊＊をFTTI以内に検知すること		B				
		理由	誤ってユーザーの＊＊＊＊動作と逆の動作をさせないようにするため。						
		説明	・＊＊＊＊判定に関しては＊＊＊の値と＊＊＊の値を使用すること。						
			＜異常判定＞			SSM-07			
	TSR	□□□	TSR 1b-1-1	ユニットBは出力信号地の異常を判定すること。 【補足】SW確定データがデータ化けにより意図しないデータとなっていないかを確認する	B		S/W		SYS-D-03　STEP5
	TSR	□□□	TSR 1b-1-2	ユニットCがCAN出力するデータの異常を判定すること。 【補足】CAN送信用データがデータ化けにより意図しないデータとなっていないかを確認する	B		S/W		SYS-D-03　STEP5
						SYSSM-01			
							H/W		すべてのシナリオにおいて、CPU内処理のすべてのSTEPに適用
									すべてのシナリオにおいて、CPU内処理のすべてのSTEPに適用

USDMの右側を活用し、ASILランクや安全方策、独立配置要求等の記述を行うことでTSRに紐づく成果物のトレーサビリティを明確する

A	B	C	
	技術安全要求（TSR）	ソフトウェア安全要件（SSR）	
TSR	TSR 3-1-2	２つのセンサの相関から故障検出を行うこと	B
	理由	単独では故障判定ができないため Part 5 Annex D Table D.11-センサー「センサーコリレーション」が典型的カバレッジ「高（99%）」であるため選択	
	説明	一定の範囲内であれば正常とする	

ANNEX Dに記述されている内容を理由欄に記述することでなぜこのSSRとなったかが直ぐに把握出来るようにする

※ANNEX Dとは規格に記述されている附属書のこと

膨大な成果物や資料との繋がりを把握出来る様な仕組みを構築
また、HSRに関してもSSR同様にUSDMを活用して対応

6.成果

上流工程への注力化

- 従来のソフトウェア開発手法では後工程まで仕様が明確にされておらず、手戻りの多い開発を行っていた。改善活動としてUMLやUSDMを活用することで上流工程に注力することが出来、後工程での不具合への低減に繋げることが出来た。
- USDMを活用することで要求と仕様を階層的に表現し、理解度の向上と曖昧さ・抜け漏れを防ぐような仕組みを構築することが出来た。
- 自動化といった工夫を行うことで工数低減ならびに人的ミスを防止出来る様な環境の構築を図れた。

機能安全への対応

- USDMを使用することでSGから各安全要求の繋がりを明確に定義することが出来た。
- USDMの右側を機能安全に特化したトレーサビリティマトリックスとして使用することで様々なドキュメント間の繋がりを表現することが出来た。

改善活動を通じてUSDMと出会い、様々な恩恵を受けることが出来た

7.今後の展開

今後の展開

①ガイドラインの更なるブラッシュアップ

- 製品機能開発と機能安全開発の有効な手段として確認することが出来たが、双方の開発整合性が高められるようにガイドラインの改善を行う
- 機能安全対応として新たに検討した内容のフィードバックを行う

②自動化の検討

- 自動化出来るところはとことん自動化させる
- 文章ライブラリを運用し、統一化された用語の自動化に向けたトライアルを行なう

③普及活動

- 今回実施して得られた成果を共有するために他メンバーへのフィードバックと普及活動の推進を行う

手戻りのない開発を目指して更なる推進をしていきたい

END

御清聴ありがとうございました