

# 第6回 D-Case研究会

## Adaptive Cruise Control(ACC) システムにおけるD-Case実証実験

株式会社 豆蔵 技術コンサルティング事業部  
井上 樹  
皆川 誠

# 自己紹介：井上 樹(いのうえ たつき)

- 2000年、豆蔵設立時に入社
- 現在は、組込系ソフトウェア開発における、オブジェクト指向・UML・SysML導入支援、モデルベースのシステムエンジニアリング・ソフトウェアエンジニアリングプロセス定義、プロセス改善の教育やコンサルティングを主に担当
- これまでに支援したドメイン：
  - 医療機器、OA機器、車載機器、精密機器、半導体製造装置、携帯電話、Web系情報システム等々、50社以上で支援
- 著書
  - 「いちばんやさしいオブジェクト指向の本 第二版」、「ダイアグラム別UML徹底活用 第二版」

# アジェンダ

- 背景
- 課題
- アプローチ
- 実施
- 結果
- 考察
- 今後の課題

# 背景

- ISO26262のセーフティケースをきっかけにアシュアランスケース(以下AC)・GSNの存在を知る
- ACは有効な技術だと思うが、開発の中でいつ書いたら良いのか分からない
  - 公開されている事例を見ると成果物が一通りできてから、事後に書いているように見える
  - 事後に書くのは規格対応としては良いが、エンジニアリングに対しては意味が無いのでは？
- エンジニアリングにおいてもっと効果的に使えるのでは？

# 課題

- ① ソフトウェアエンジニアリングから見たときACにどのような効果があるのか？
- ② ACを作る効果的なタイミングは？

# アプローチ(1)

## • 課題①「ACの効果」に対して

– ソフトウェア開発内で行われる議論を対象とし、ACの適用に効果があるかを観察

- 対象とする議論としては「品質保証」を選択
- ACの効果として以下の仮説を立てた

QCDのうちQの議論が身近なため

一般的な品質の観点

一般的なACの効果

	プロセス品質	プロダクト品質
開発者間コミュニケーションのサポート	プロセステラリングで決定された実施内容に関する、開発者間での合意形成の支援	成果物に対するチェックリストの内容理解の支援
主張が達成されていることの証明	所定のプロセスが実施されたことの証明	各成果物が定められた品質を満たしているかの証明

# アプローチ(2)

## • 課題②「AC作成のタイミング」に対して

- 各工程のプロセスを決める際にACの作成を加味
- 基本的な考え方
  - 各工程において開始時や終了時のように開発者間での議論や活動結果の確認が必要となる部分でACをからめた活動を行う

成果物としてどのような内容のものを作るのかを確認

ACによるクライテリアの事前チェック

その工程の活動

成果物が定められた品質を満たしているかを確認

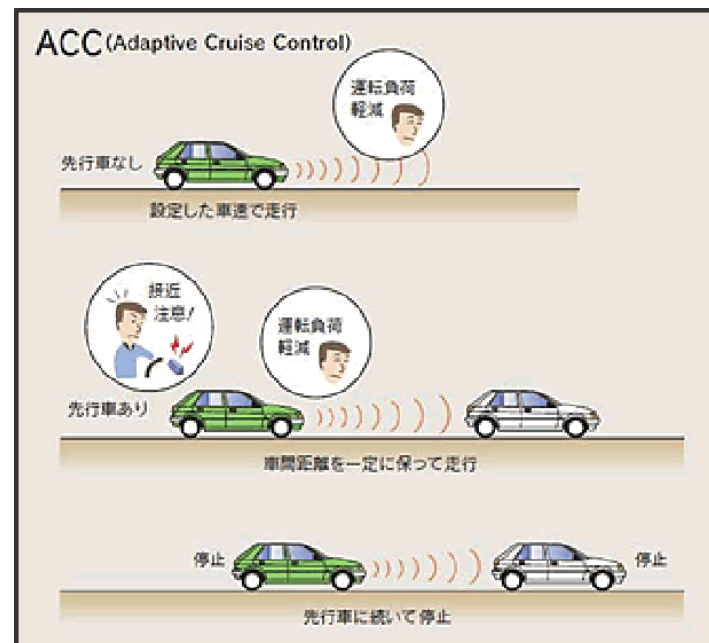
ACによるクライテリアの事後チェック

# アプローチ(3)

## • 実際にもものを作っているしながら試行

### – 題材: Adaptive Cruise Control (ACC) を選択

- 題材として様々な研究で使われているため、情報が入手しやすく、ISO可された要件 (ISO15622) もある
- 以前にも題材として選択したことがあり、モデル化の知見がある
  - AC導入前後の比較もできる

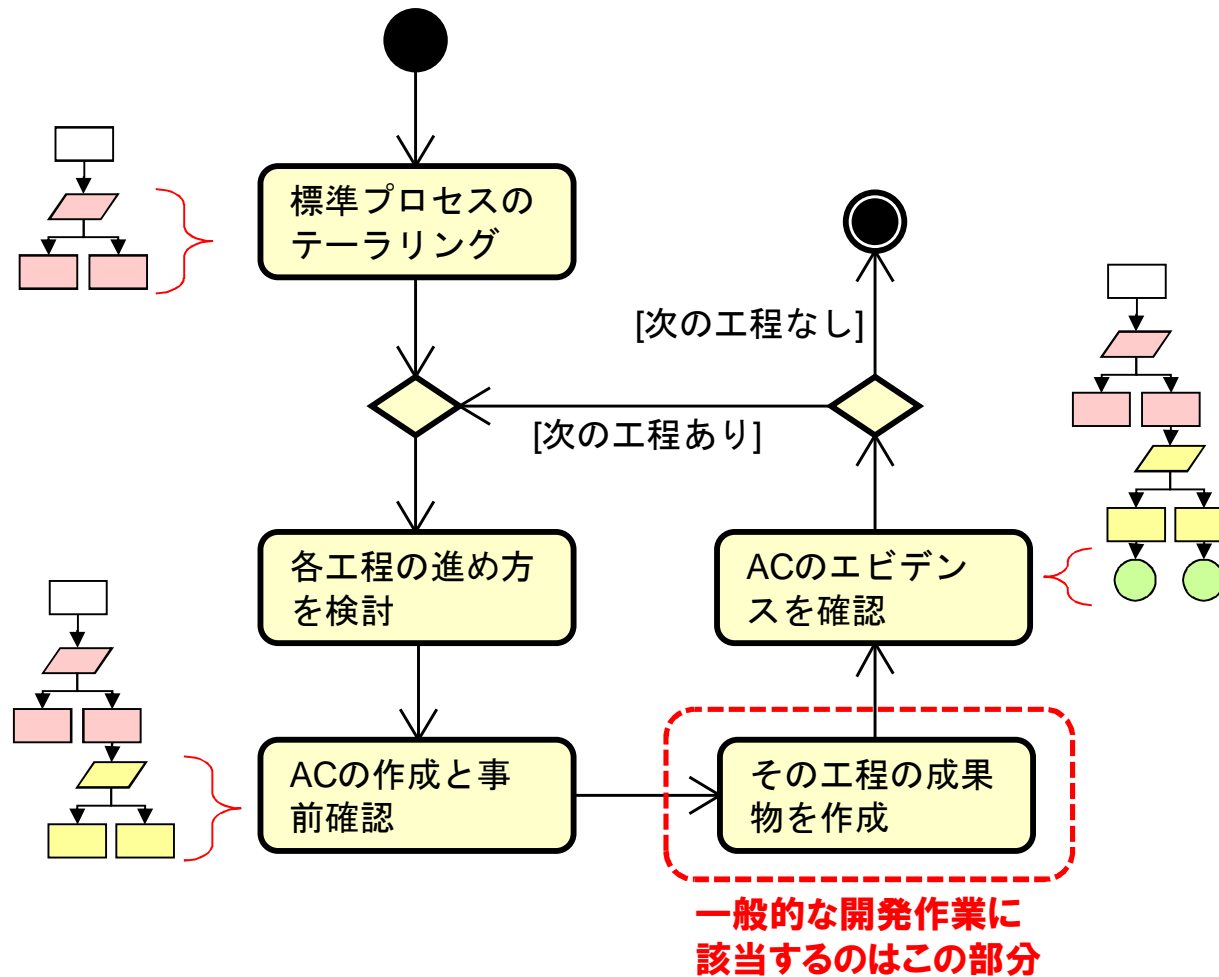


国土交通省 先進安全自動車のサイトより  
(<http://www.mlit.go.jp/jidosha/anzen/01asv/japanese/practical.html>)



# 実施(1)

- ACCを題材に以下の活動を実施した



# 実施(2)

- 期間:7ヶ月、体制:2名(後半3ヶ月は1名)
- 工数:2人月(40人日)
- 成果物

工程	成果物
要求分析	ユースケース(ユースケース図、ユースケース記述、シナリオリスト) ドメインモデル(クラス図、ステートマシン図、シーケンス図) リスク分析(FTA図、HAZOPシート)
要件作成	ステートマシン図
アーキテクチャ設計	アーキテクチャ要件リスト、アーキテクチャの静的構造(クラス図)、メカニズムの設計結果
詳細設計	設計課題リスト、ソフトウェア構造のクラス図、クラス図の動作確認(シーケンス図)
実装	Java(約7000行(コメント込)、約3600行(コメント抜))
テスト	システムテストケース、システムテスト結果

# 結果(1)

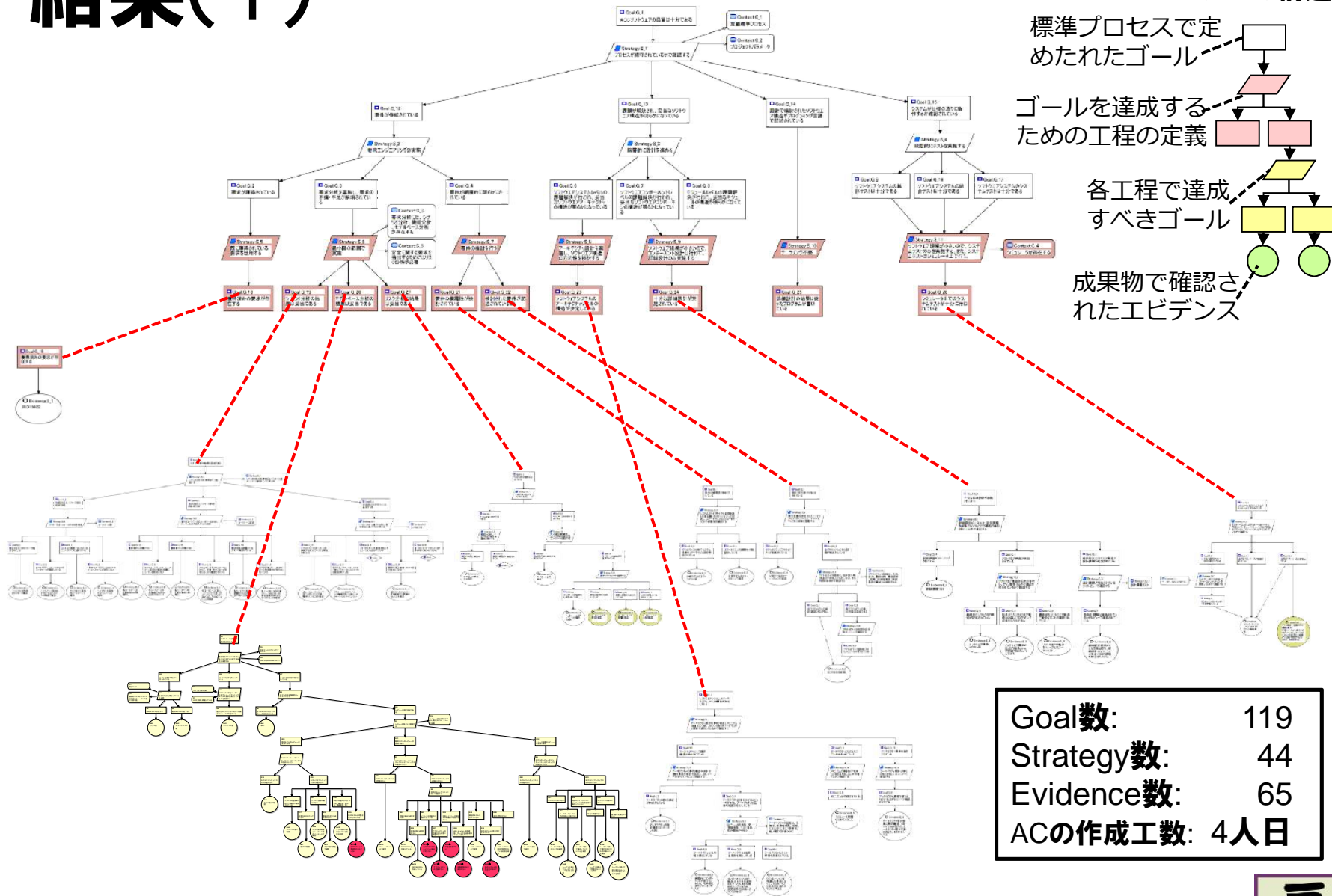
## このACの構造

標準プロセスで定めたゴール

ゴールを達成するための工程の定義

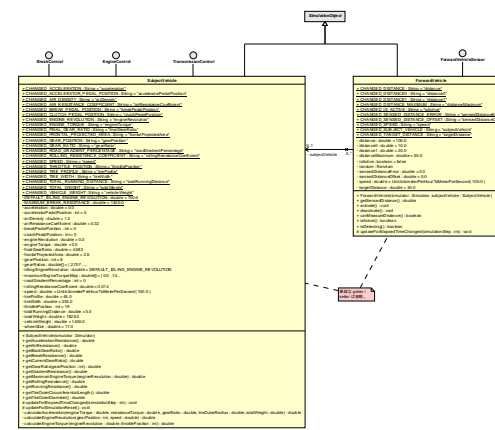
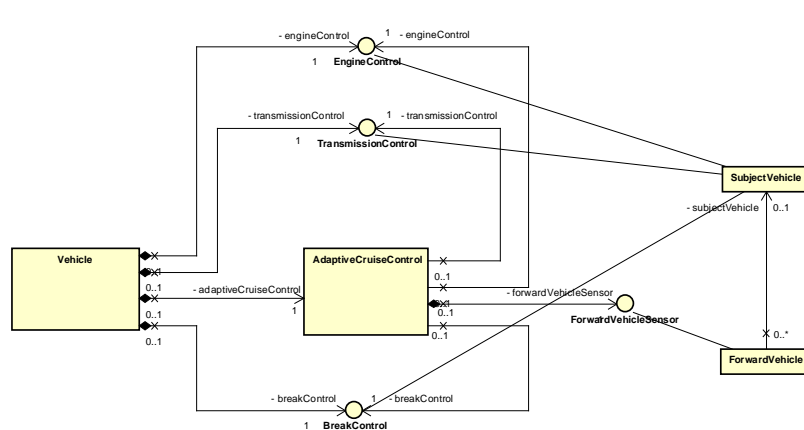
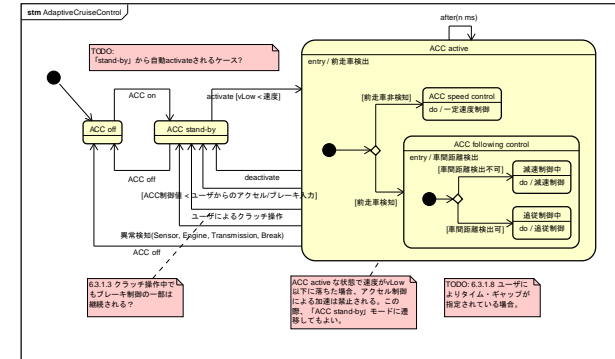
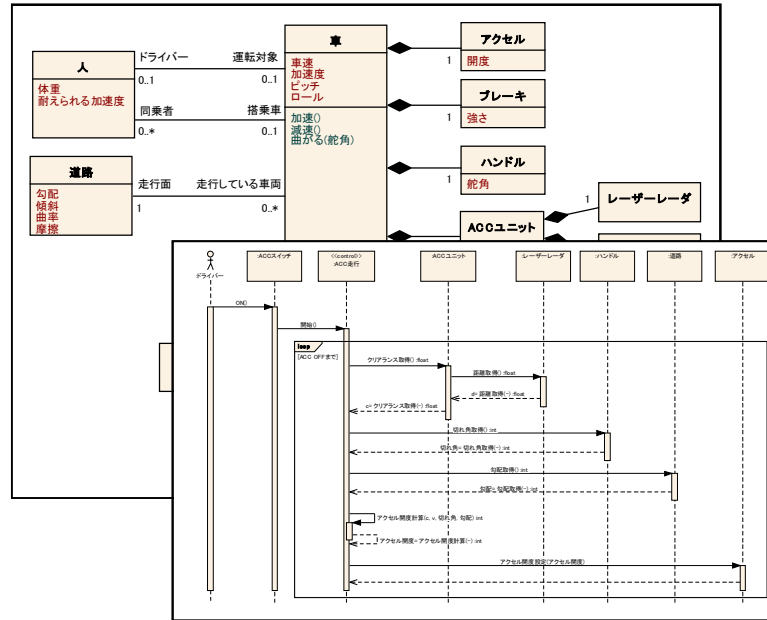
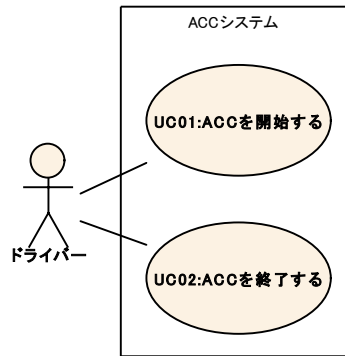
各工程で達成すべきゴール

成果物で確認されたエビデンス



# 結果(2)

## • 作られたモデル



# 結果(3)

## ・動作シミュレーション

The screenshot displays a simulation environment with a car on a road. The interface is divided into several sections:

- クラッチ ブレーキ アクセル**: Three vertical gauges showing clutch, brake, and accelerator pedal positions, all at 0%.
- 白車**: Vehicle parameters table.

走行距離[m]:	5411.6	適用
速度[km/h]:	91.4	適用
加速度[m/s^2]:	-0.310	適用
エンジン回転数[rpm]:	2394.7	適用
エンジン軸トルク[N・m]:	0.000	適用
スロットル開度[%]:	0	適用
車両重量[kg]:	1825.0	適用
車両質量[kg]:	1400.0	適用
タイヤ幅[mm]:	235.0	適用
タイヤ扁平率:	45.0	適用
ホイールサイズ[inch]:	17.0	適用
- トランスミッション**: Gear and shift parameters table.

ギア・ポジション:	6	適用
変速比(後退):	2.707	適用
変速比(1速):	2.909	適用
変速比(2速):	1.944	適用
変速比(3速):	1.434	適用
変速比(4速):	1.100	適用
変速比(5速):	0.868	適用
変速比(6速):		
最終流速比:		
- 走行抵抗**: Resistance parameters table.

総走行抵抗[kgf]:	57.733	
空気抵抗[kgf]:	31.570	
空気密度[kg/m^3]:	1200	適用
空気抵抗係数[kg・s^2/m^4]:	0.320	適用
前面投影面積[m^2]:	2.500	適用
ころがり抵抗[kgf]:	25.550	
ころがり抵抗係数:	0.014	適用
勾配抵抗[kgf]:	0.000	
- 先車**: Lead vehicle parameters table.

先車あり:	<input checked="" type="checkbox"/>	
速度[km/h]:	90.0	適用
車間距離[m]:	44.447	適用
測距誤差(+)[m]:	0.0	適用
測距オフセット[m]:	0.0	適用
測距計測値[m]:	0.0	
- シミュレータ**: Simulation controls.

経過時間:	00:04:32.520	
更新間隔[ms]:	10	適用
実時間調整:	<input checked="" type="checkbox"/>	
Buttons:	start stop stop reset	
- AdaptiveCruiseControlFrame**: Adaptive Cruise Control (ACC) control panel.

制御開始間隔[ms]:	20	適用
システム状態:	オフ <b>スタンバイ</b> 巡航制御中 追従制御中 故障中	
目標巡航速度[km/h]:	100.0	適用
設定可能な最大巡航速度[km/h]:	120.0	適用
設定可能な最小巡航速度[km/h]:	60.0	適用
加減速制御可能な最低速度[km/h]:	40.0	適用
目標タイム・ギャップ[ms]:	2000	適用
設定可能な最大タイム・ギャップ[ms]:	2200	適用
設定可能な最小タイム・ギャップ[ms]:	1500	適用
現在の走行速度[km/h]:	99.1	
Buttons:	オン オフ 制御開始 制御休止	

# 考察(1)

## • 課題①アシュアランスケースの導入効果について

	プロセス品質	プロダクト品質
開発者間コミュニケーションのサポート	テーラリング時に標準的な手順からどこを割愛するか、割愛しても良いかをACを見ながら議論することができた。	成果物作成時にACを見ながら成果物のどの辺りに力を入れて作ればよいかを確認することができた。
主張が達成されていることの証明	テーラリングの検討内容がACとして残っていることで、成果物の不具合がテーラリングのミスに起因することを発見することができた。	成果物のチェックをACを見ながら行うことで、問題の抽出、及び、それが本当に問題なのかを確認することができた(次頁参照)

今回の試行でACの導入効果を最も感じた部分

# 考察(2)

成果物	抽出数	抽出できた問題
ユースケース図	0	なし
ユースケース記述	2	<ul style="list-style-type: none"> <li>● UC01の「課題やTBD」が未解決</li> <li>● UC01の例外フロー1bの発生条件が不明確。</li> </ul>
シナリオリスト	2	<ul style="list-style-type: none"> <li>● 荒唐無稽なシナリオが挙げられていない(※)</li> <li>● シナリオのパラメータが具体的に考えられていない。</li> </ul>
ドメインモデル	6	<ul style="list-style-type: none"> <li>● 他のモデルとの整合が取れていない(クラス図)</li> <li>● イベント抽出が足りない(ステートマシン図)</li> <li>● アクティビティの定義がない(ステートマシン図)</li> <li>● 状態遷移表がない(ステートマシン図)</li> <li>● 自動遷移の条件がない(ステートマシン図)</li> <li>● 他のモデルとの整合が取れていない(ステートマシン図)</li> </ul>
リスク分析結果	5	<ul style="list-style-type: none"> <li>● FTAの外的(環境)要因の抽出が足りない(豪雨、吹雪等)</li> <li>● FTAのオペレータ要因の抽出が足りない(ACCスイッチ誤操作)</li> <li>● HAZOPの「考えられる要因」に「(省略)」が存在(※)</li> <li>● HAZOPの「起こりうる影響・結果」に「(省略)」が存在(※)</li> <li>● HAZOPの「対策」に「(省略)」が存在(※)</li> </ul>
システム要件	0	なし
アーキテクチャ設計モデル	0	なし
詳細設計モデル	0	なし
ソフトウェアシステムテスト仕様書	1	<ul style="list-style-type: none"> <li>● OKになっていないテストケースが含まれている</li> </ul>



# 考察(3)

## • 課題②作成タイミングについて

- 各工程の事前・事後にACの作成・確認を行うことによる効果は見られた
- 他のタイミングで作成した場合にどのような効果が得られるのかは今後の課題

## • その他の考察

- ゴールとして、活動名(ex. 設計を実施していること)を書いても意味が無い
  - 活動の目的レベルで記述しないと意味が無い
    - ex. 全ての設計課題に対し、解決策が選定されていること
- AC作成時に目的を明確にしておかないと、作成中にゴール分解の方向性が変わってしまう
- 対象に対する理解度に応じてゴール分解の細かさが変わってしまう
- 「活動を行わなかったこと」のエビデンスはどう記述するのか？



# 今後の課題

## • 導入プロセスの効果測定

- 今回はほとんどの成果物で成果物の作成者がACの作成にも関わっていた
  - ゴールの達成ではなく、成果物の作成負荷という意識が働いた状態でACを作っている可能性が高い
- より実践的な開発では成果物の作成者とACの作成者は別人であると考えられるため、その点から見ると今回のような成果が出るかは分からない

## • プロセスアシュアランスのためのパターン

- ACを作成していくに従い、作成時間が短くなっていったので、何らかの思考のパターンが存在していると考えられる

# さいごに

- 今回の実証実験は電気通信大学の松野先生のご支援の下で実施いたしました。
- 改めてお礼申し上げます。