

The DENSO logo is written in a bold, italicized, red sans-serif font.

Crafting the Core

プロセス分析表による プロセス改善方法の最適化の提案

(株) デンソー

AD&ADASシステム技術部
岡 和道

コアスキル開発部
古畑 慶次

目次

1. 担当製品
2. 課題
3. 解決策の提案
 - ・原因箇所特定の容易化
 - ・対策の容易化
4. 検証
5. まとめ

1. 担当製品 (1)

■ デンソーの紹介

● 会社概要



所在地：愛知県刈谷市

従業員：単独 45,304人

：連結 171,992人

● 事業分野

○自動車関連

—<AD&ADAS事業部>

運転支援製品の開発設計

ミリ波レーダ
カメラセンサ
レーザレーダ
周辺監視
ロケータ

私の所属部署

○生活関連機器

○産業機器

○新事業

1. 担当製品 (2)

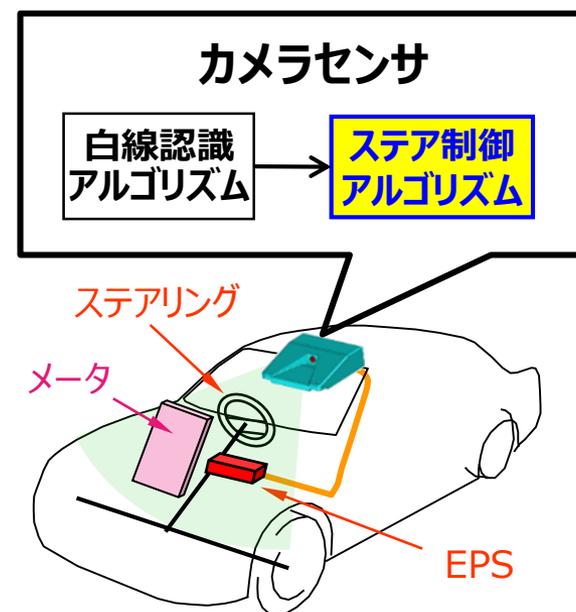
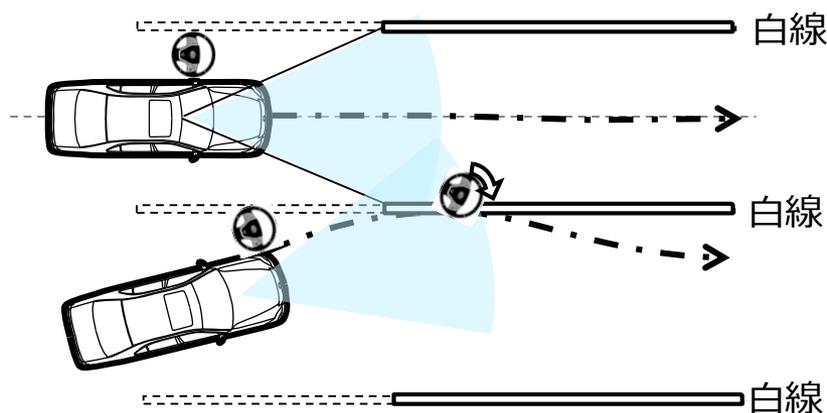
■ 私の担当業務

- 運転支援システムのステアリング制御のソフトウェア開発

■ ステアリング制御の機能

- カメラからの情報を元にステアリングを制御する機能

例) レーンキープなど

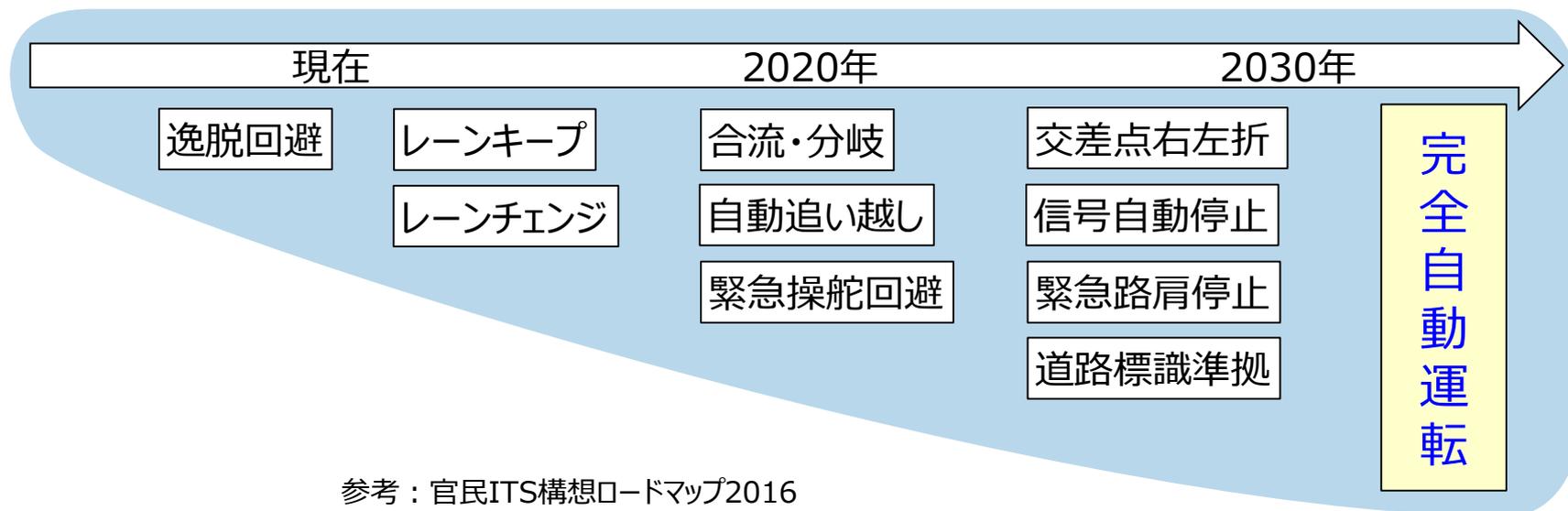


ステアリング制御で安全/利便機能を実現している

1. 担当製品 (3)

■ 業界動向

- ステア制御は車両の曲がる機能の要であり、必要不可欠である。
- 自動運転は高速道路から一般道に拡大しており、対応が必要なシーンも増加が予想される

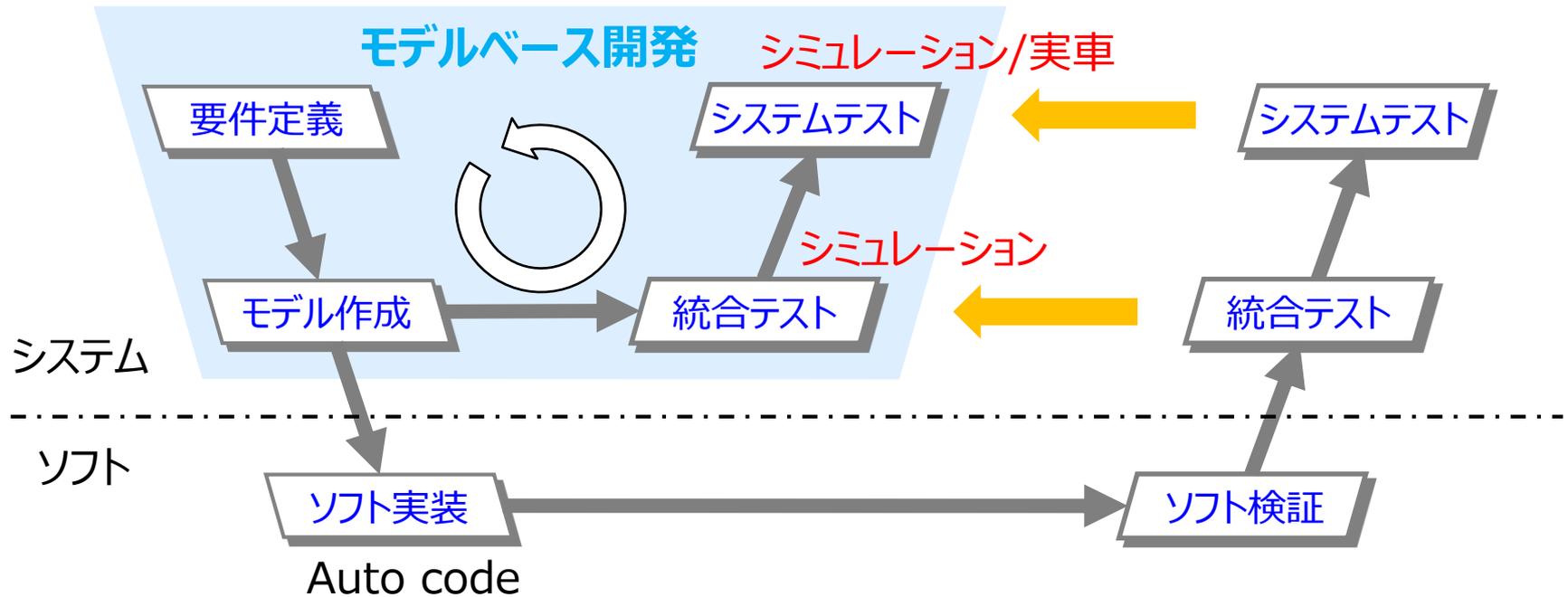


自動運転に向けて運転支援システムは急速に多機能化していく

2. 課題 (1)

■ 開発手法

- 開発期間短縮を目的としてモデルベース開発を採用



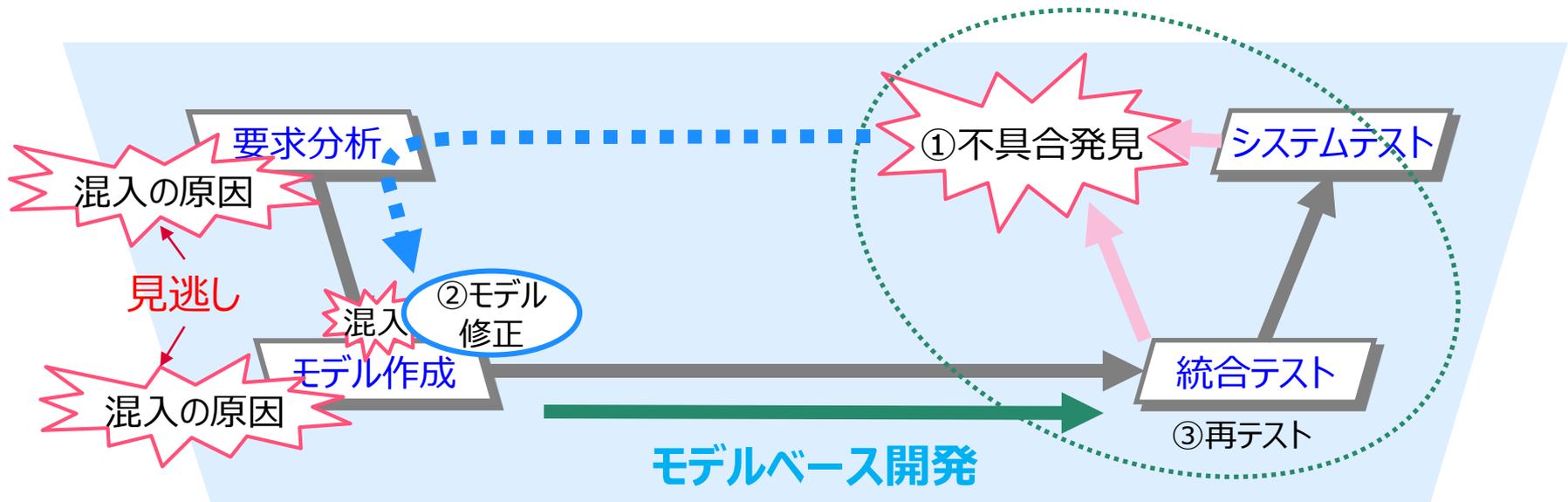
シミュレーションによる仕様設計の高速化が可能

2. 課題 (2)

■ 高速シミュレーションの罠 ①

不具合発生時に容易なモデル修正のみを実施し、開発時間が増大する罠

- モデルが修正されれば不具合は解消する（流出防止は可能）
- モデルの修正結果はシミュレーションで容易に検証可能
- モデルを直接修正し不具合混入の根本原因の対策までは実施されない

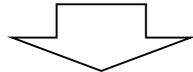


混入の根本原因が見逃される可能性有り

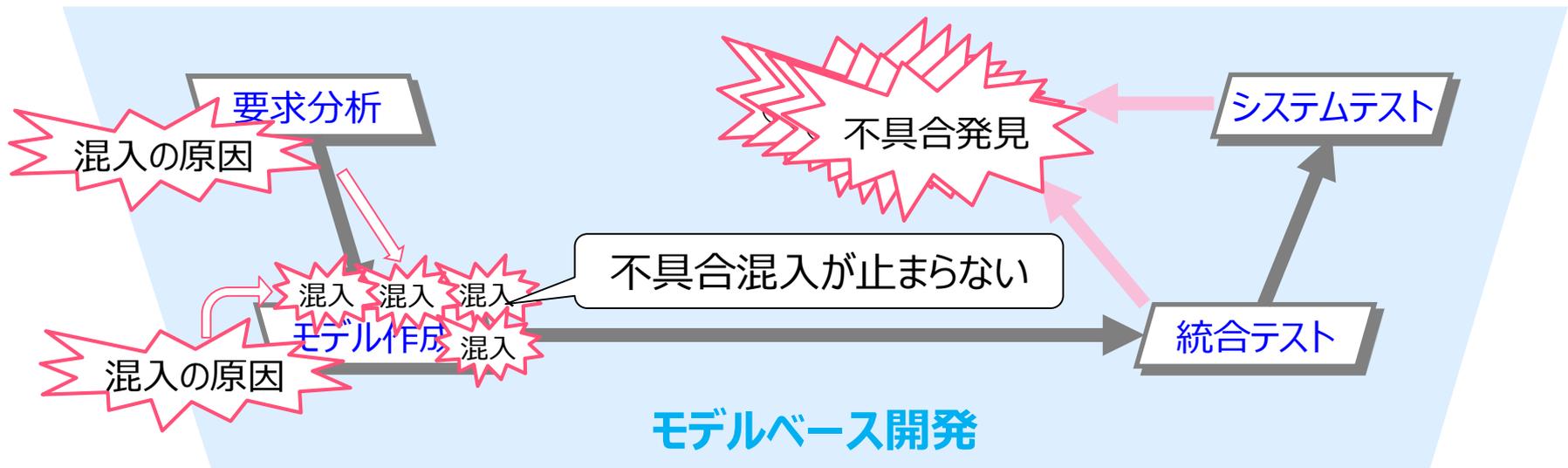
2. 課題 (3)

■ 高速シミュレーションの罠②

- モデルでの修正により混入の根本原因への対応が見逃される



- 開発規模拡大時に同要因での不具合が増大する
- 修正工数が増大する（モデルベース開発の導入意図に反する）



混入の根本原因を対策する必要がある

2. 課題 (4)

■ 高速シミュレーションの罠に陥る理由

理由1) 混入の根本原因まで対策すると時間がかかる

- なぜなぜ分析等での根本原因の特定
- 対策の考案と決定
- 対策の実施



時間がかかる

理由2) 混入の根本原因での不具合対策には経験とスキルが必要

- なぜなぜ分析で正しい根本原因の導出
- 自身の得手不得手に影響されない対策の判断
- 対策の正確な実行

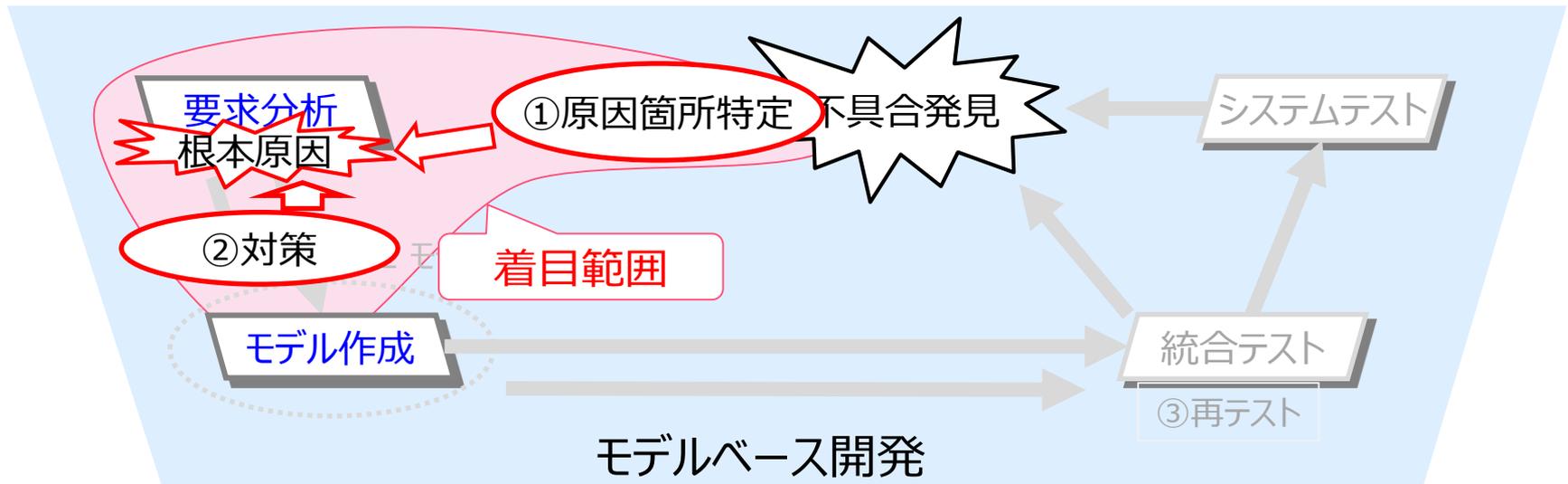


難易度が高い

工数負荷も難易度も高い根本原因への対策は避けられてしまう

2. 課題 (5)

- 課題の定義：混入の根本原因に対策する作業を容易化する
 - 「原因箇所特定の容易化」(①)
 - 「対策の容易化」(②)



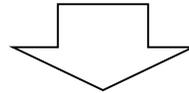
「原因箇所特定の容易化」と「対策の容易化」を解決する事で
高速シミュレーションの罣を回避する

3. 解決策の提案

■ 解決策の全体像

● 「原因箇所特定の容易化」

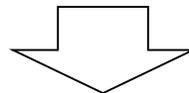
- 不具合の発見箇所と原因箇所を短時間で合理的に対応させる方法が必要。



PFD (Process Flow Diagram) を活用

● 「対策の容易化」

- 原因箇所に対処可能な対策候補を短時間で理論的に抽出し、その候補から合理的に最適解を導き出す方法が必要。



プロセス分析表を提案

3.1 原因箇所特定の容易化（1）

■ コンセプト

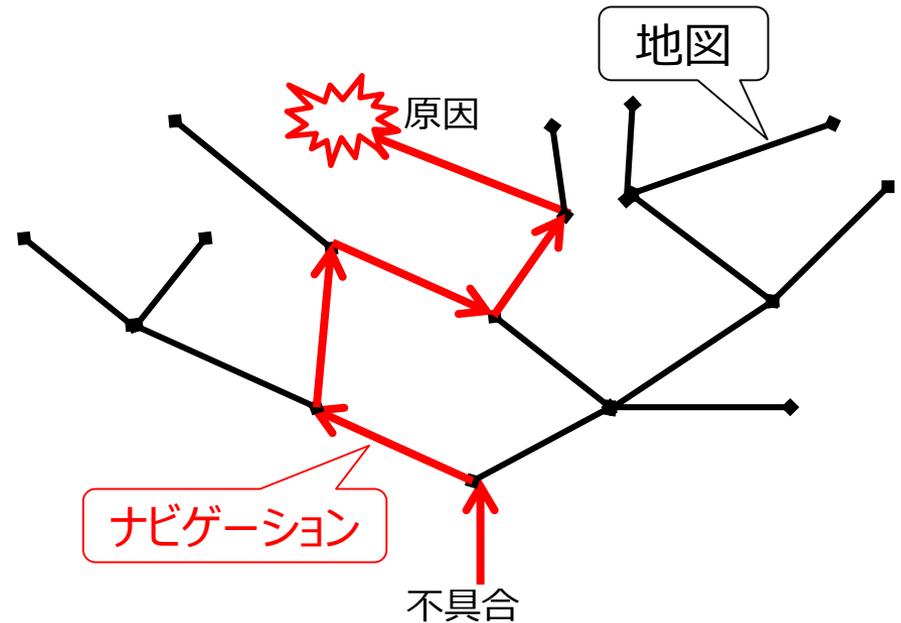
- 不具合の原因箇所に遡るための地図を用意しナビゲーションする

「地図」

開発プロセスで実施する作業をあらかじめ地図にしたもの

「ナビゲーション」

開発プロセスの地図情報から、不具合をスタートとし、原因箇所をゴールとした方向を示す

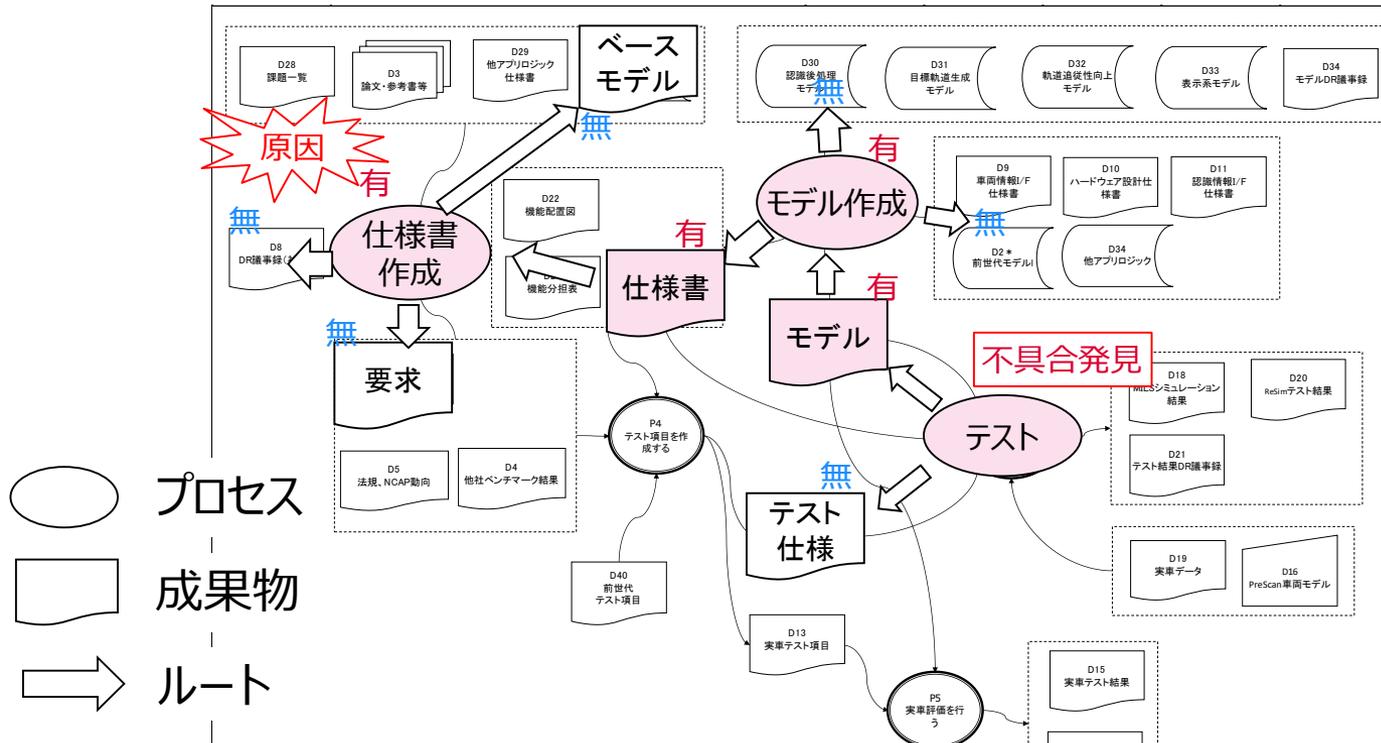


地図上に不具合から原因箇所に戻るまでのルートを示す

3.1 原因箇所特定の容易化（2）

■ 実現方法

- 地図⇒PFDを活用
- ナビゲーション⇒PFD上で不具合と原因を合理的に結ぶ



PFDにより、不具合混入の原因を迅速に特定可能

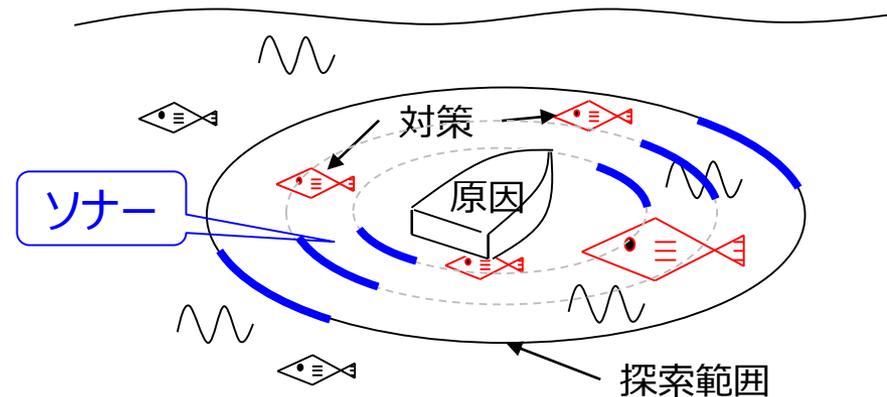
3.2 対策の容易化（1）

■ コンセプト

- 原因箇所への対策候補を探索し大きさ（効果）を比較するソナーを用意する

「ソナー」

- ・規定した範囲で対策の全方位探索を実施する
- ・発見した対策候補の効果を示す



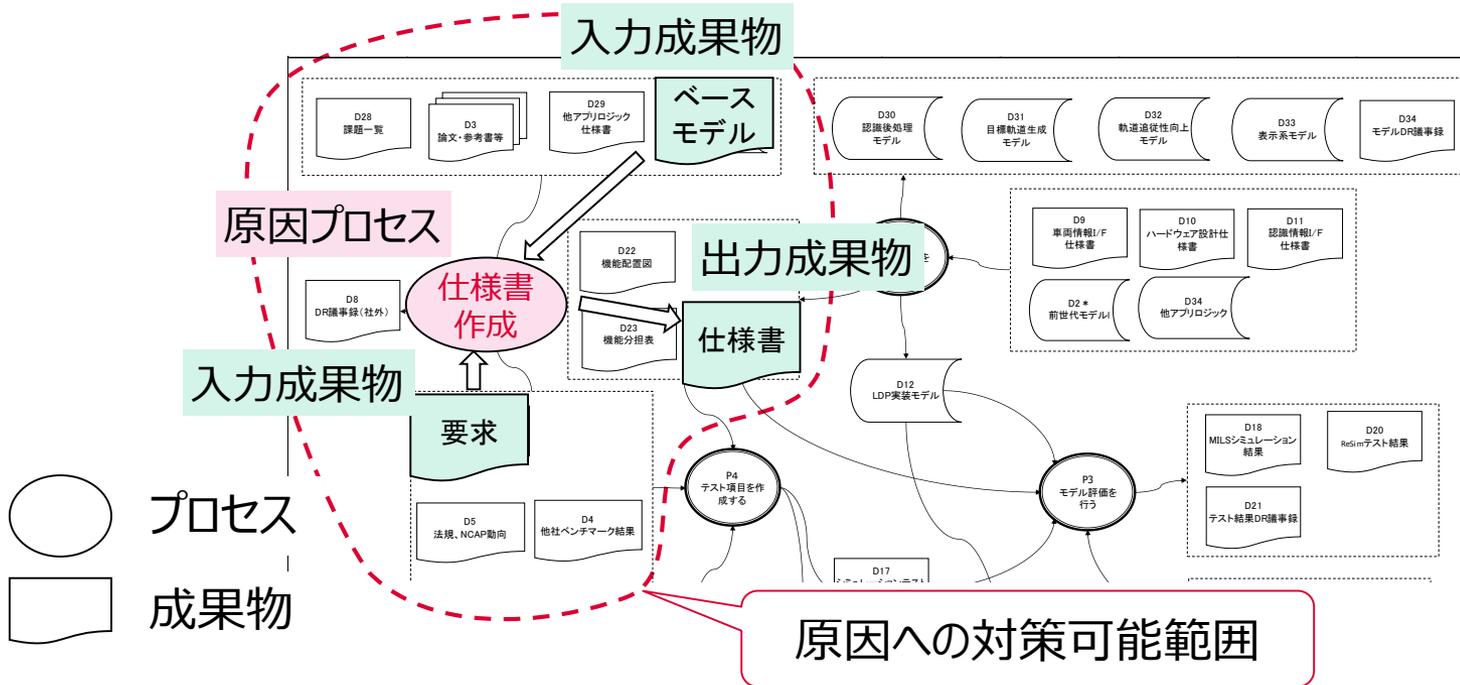
対策候補の探索と比較を可能にする

3.2 対策の容易化 (2)

■ 実現方法 ①

- PFDにより対策可能範囲を特定する

PFD上の**原因プロセス**とその**入出力成果物**を対象範囲とする



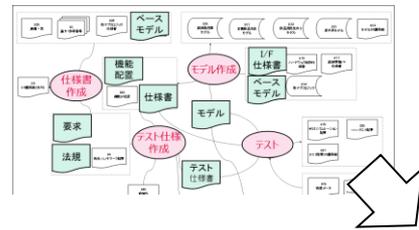
PFDにより、対策可能範囲を特定可能

3.2 対策の容易化 (3)

■ 実現方法 ②

- プロセス分析表により効果最大の対策を選択

プロセス分析表：PFDでは比較しづらい対策候補の内容を表形式で可能にしたもの



問題			原因		プロセス番号						成果物番号							
NO	登録日	登録者	問題 (不具合の状況)	原因	重要性	P1.8.1	P1.8.2	P1.8.3	P2.4	P2.6	D1	D2	D5	D12	D29	D39	D40	D41
1	16/8/22	H			A		○				○						○	

原因プロセスでの対策

入力成果物での対策

出力成果物での対策

プロセス分析表により、対策内容を比較可能

3.2 対策の容易化（4）

■ プロセス分析表の有効性

行（不具合）が増加した際に同一対策で色分けすると

- 他の不具合にも効果が得られる効率的な対策が選択可能になる
- 不具合が頻発している原因に対策が可能になる

NO	登録日	登録者	問題（不具合の状況）	原因	重要度	P1.8.1	P1.8.2	P1.8.3	P2.4	P2.6	D1	対策	D2	D5	D12	D29	D39	D40	対策	D41
1	'16/8/2 2	H			A	○					○	対策②						○		
2	'16/8/2 3	H																		○
3	'16/8/2 3	H																		○
4	'16/8/2 9	H			B															○
5	'16/8/3 1	H			C															○
6	'16/9/1 9	H			A						○	対策①								○
9	'16/9/2 1	H			B															○
10	'16/10/ 29	H			A						○									○

効率的な対策の組み合わせが可能になる

複数の不具合を発生させる原因に対策可能

対策①

対策②

対策③

対策④

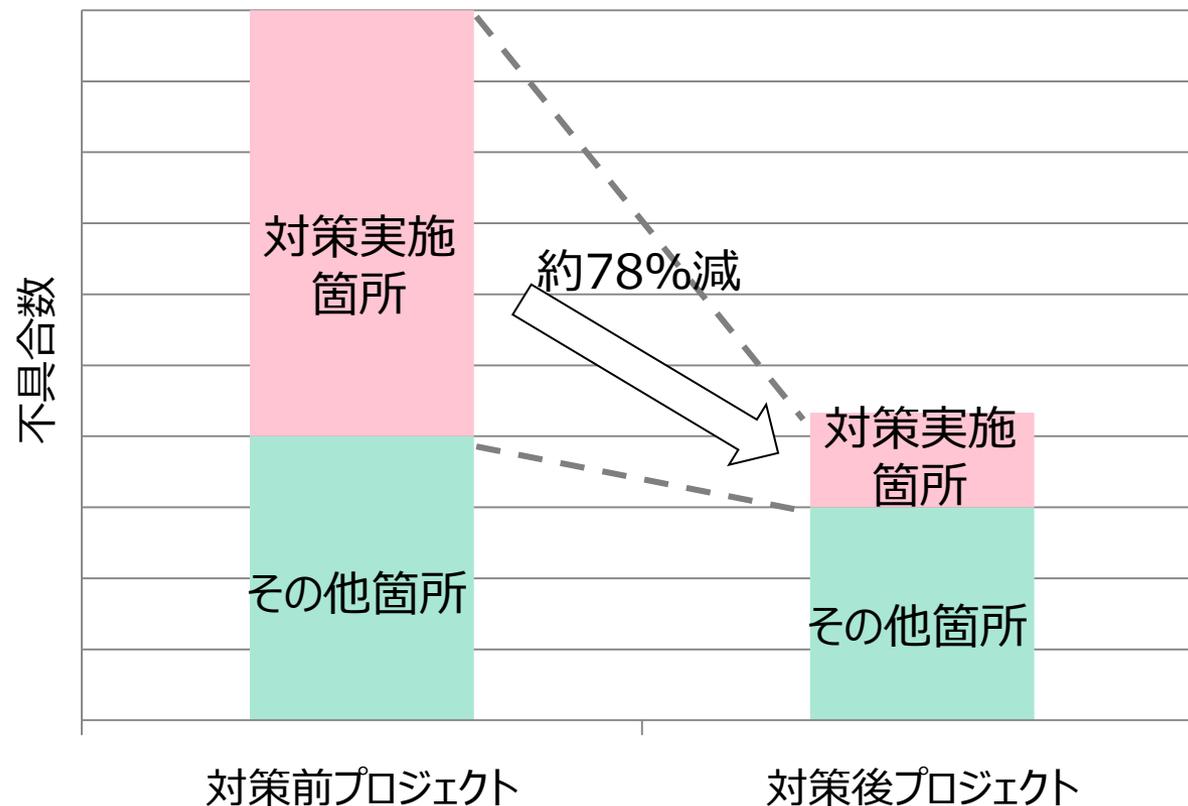
4. 検証（1）

■ 検証方法

- 提案するPFDとプロセス分析表を活用した不具合対策の効果を検証するため、以下の条件で調査を行った。
 - 提案手法を実施前（モデル修正のみ実施）と実施後（モデル修正＋根本原因の修正）で比較
 - テストで発見された不具合件数（修正ループ回数）で比較
 - 比較する開発は同規模とする
- PFDとプロセス分析表による効果を具体的に比較するため、不具合数を以下2種類に分類しそれぞれの不具合数推移を検証する。
 - 再発が多く、根本原因の対策を実施した「対策実施箇所」
 - 単発の発生で対策の必要なしと判断された「その他箇所」

4. 検証 (2)

■ 検証結果-不具合数の比較-



対策実施箇所を要因とする不具合件数が大幅に減少した

4. 検証 (3)

■ 検証結果の考察 ①

- 対策実施箇所を要因とする不具合件数の減少は、提案する「原因箇所の特定」と「対策」の実施で、再発が発生しなくなった効果と考えられる。
- 不具合件数が全体で約60%減少した事で、修正ループに立ち戻る工数が大きく減少した。今後対策を継続する事で、モデル上の手戻りを更に減少させる事が可能と考えられる。
- 今回の検証はモデルベース開発で実施したが、対策決定手法はモデルに依存しないことから、一般的な開発にも適用可能と考えられる。

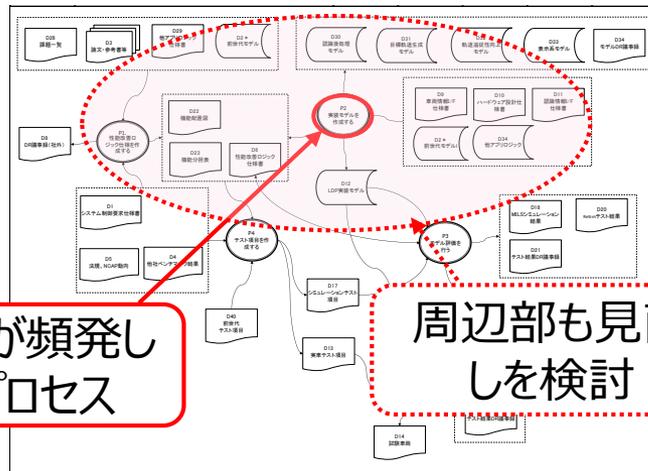
4. 検証 (4)

■ 検証結果の考察 ② -プロセス全体の最適化への活用-

- 今回作成したPFDとプロセス分析表は、不具合データを積み重ねる事で、問題の頻発するプロセス & 成果物を特定する事ができる。

⇒次期開発を始める時にプロセス全体の見直し※にも活用できる

※プロセスのレイアウトの変更・追加など



N	登録日	登録者	問題 (不具合の状況)	原因	重要度	対策											
1	'16/8/22	H			A												
2	'16/8/23	H			A												
3	'16/8/23	H			B												
4	'16/8/29	H			B												
5	'16/8/31	H			C												
6	'16/9/15	H			A												
7	'16/9/21	H			A												
8	'16/9/21	H			A												
9	'16/9/21	H			B												
10	'16/10/29	H			A												

問題の頻発箇所

次の開発時にプロセスのハザードマップになる。
(問題の発生しやすいプロセスの見える化)

5. まとめ

■ 活動のまとめ

- 要求分析での根本原因に対して「原因箇所特定」「対策」の二つを容易化する手法として、PFDの活用とプロセス分析表を提案した。
 - ⇒PFDとプロセス分析表の活用によりモデルベース開発の現場で生じている高速シミュレーションの罣を回避可能とした
 - ⇒PFDとプロセス改善表を用いてプロセス上のハザードマップを作成したことにより、次期開発時の注意箇所を見える化した

■ 今後の進め方

- 対策を継続する事で、モデル上の手戻りを更に減少可能なことを確認する。
- PFDとプロセス分析表を使用した不具合への対策手法をプロセス全体の最適化を実施する際にも活用し、効果を検証する。

End