機能安全開発現場で見えた
テンプレ遵守する初心者が陥りやすい設計の改善

「説明できない設計書」から「安全性が説明できる設計書」へ

2019/10/09

パナソニック アドバンストテクノロジー（株）

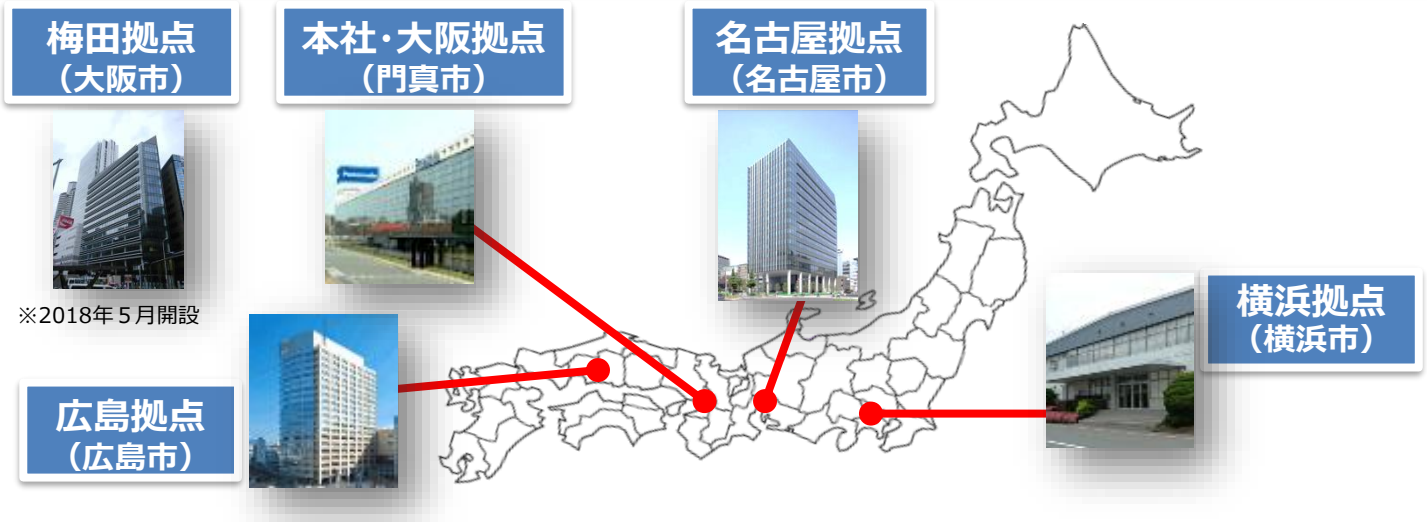
田崎 功一, 尾関 英克

発表内容

自己紹介	背景	現場の問題	事例	効果	最後に
・弊社の紹介 ・弊社の事業活動 ・PAD機能安全推進課の役割	・組織における機能安全対応 ・開発プロジェクト	・何が悪いの？ ・現場が陥っていた問題(支援前) ・現場が陥っていた問題(指導の結果) ・初期の指導の結果で分かったこと	・事例① ・事例② ・事例③ ・事例④	・改善の効果 ・改善の定量的効果	・まとめ ・今回の取り組み分かったこと ・今後の抱負

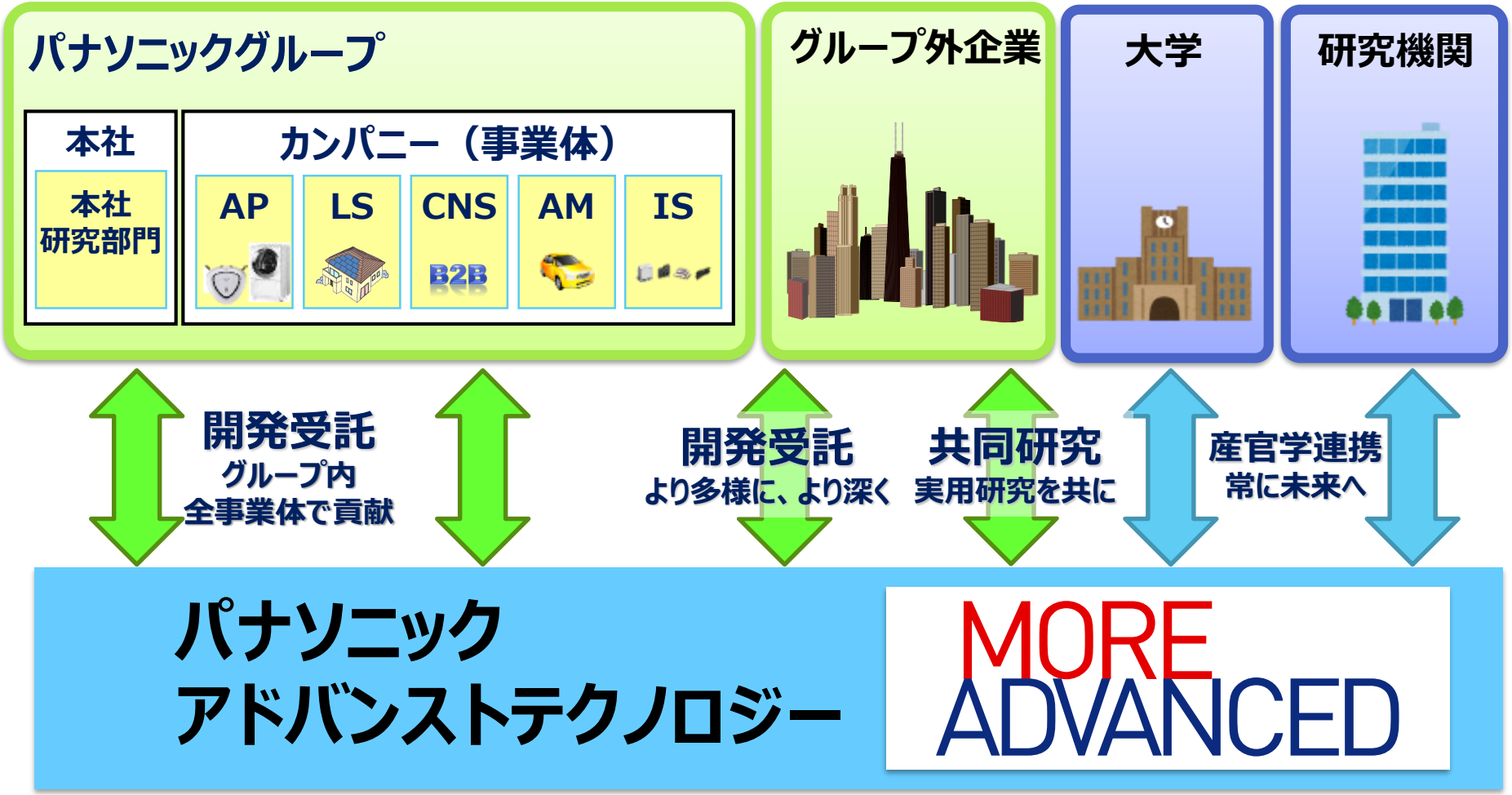
弊社の紹介

会社名	パナソニック アドバンストテクノロジー株式会社 (英文) Panasonic Advanced Technology Development Co., Ltd
設　立	2007年4月1日　(創業1985年)
事業目的	システムおよびソフトウェア設計開発を通じて、安全・安心、 快適・便利な暮らしを実現する
売上高	112億円
従業員	502名　（2019年5月1日現在）
所在地	



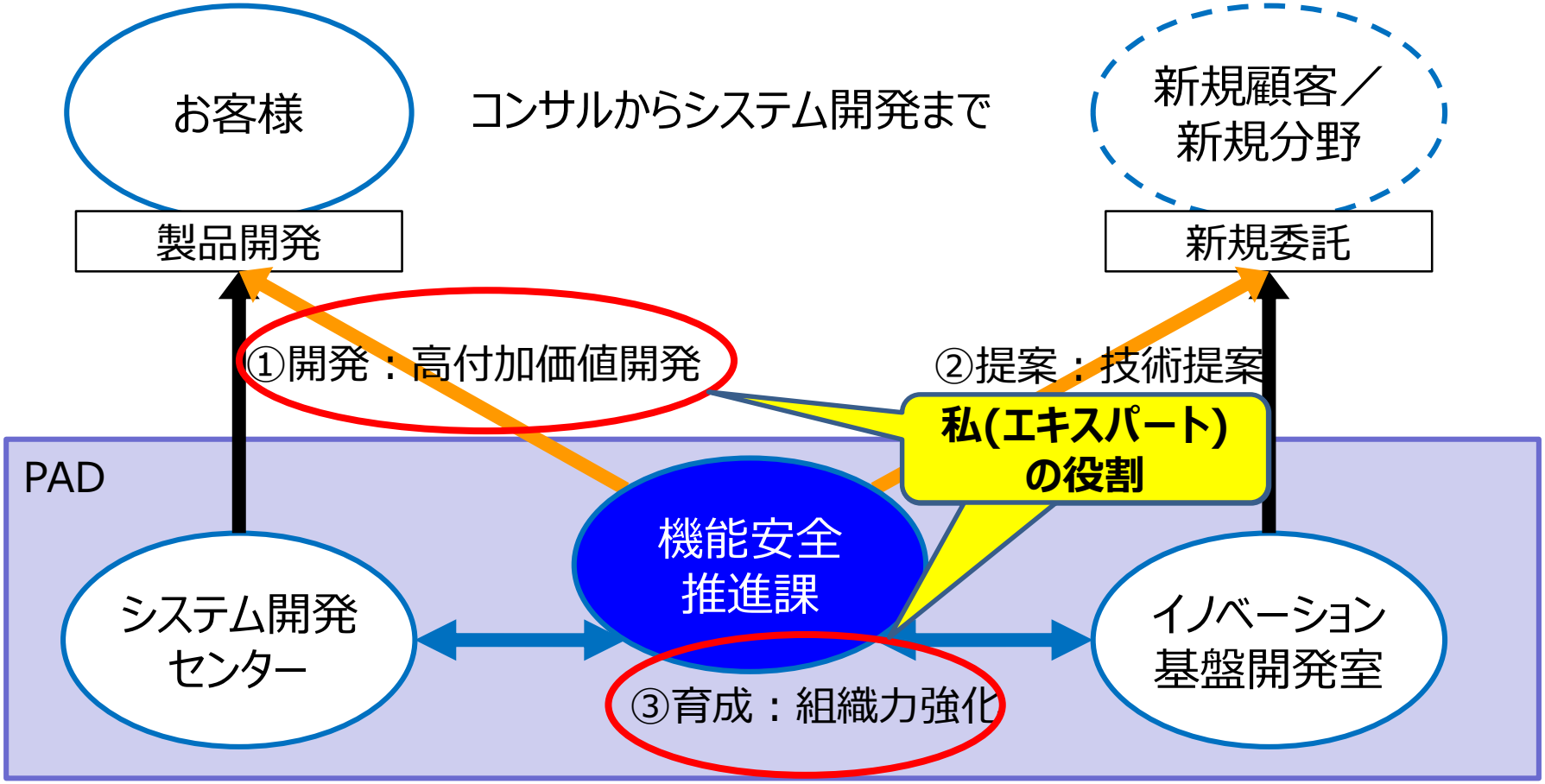
弊社の事業活動

- パナソニックグループ内にとどまらずグループ外の多様な事業分野へ範囲拡大
- 自らの意思を持って先端技術の探索・仕込みと手の内化に常にチャレンジ



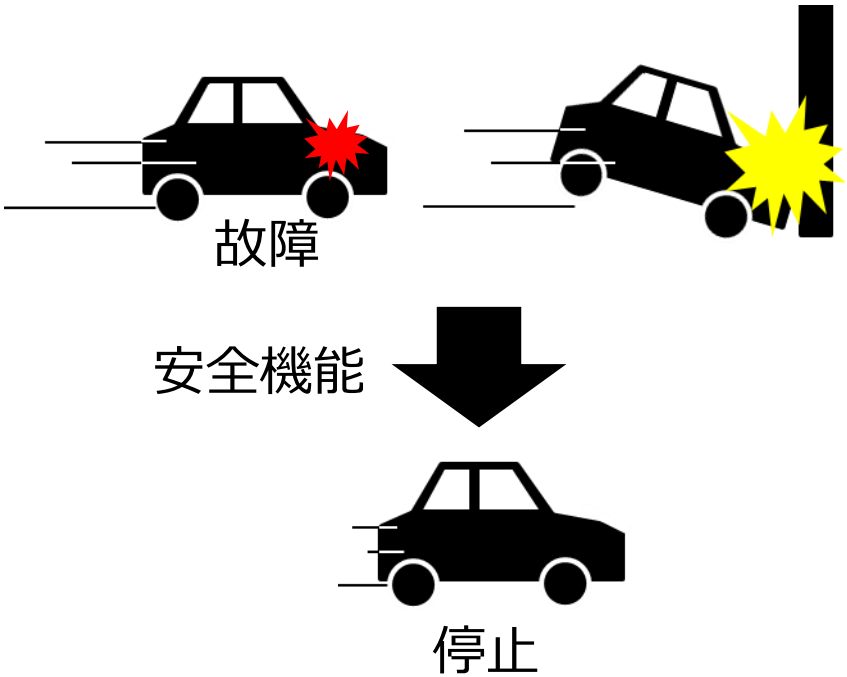
PAD機能安全推進課の役割

①開発：他社を圧倒する機能安全スキルを活かし、顧客に高い価値を提供する
②提案：高信頼分野の新規顧客に技術提案を行い、新たな委託に繋げる
③育成：機能安全関連の技術やノウハウを展開し、ＰＡＤの組織力を向上させる



背景～組織における機能安全対応～

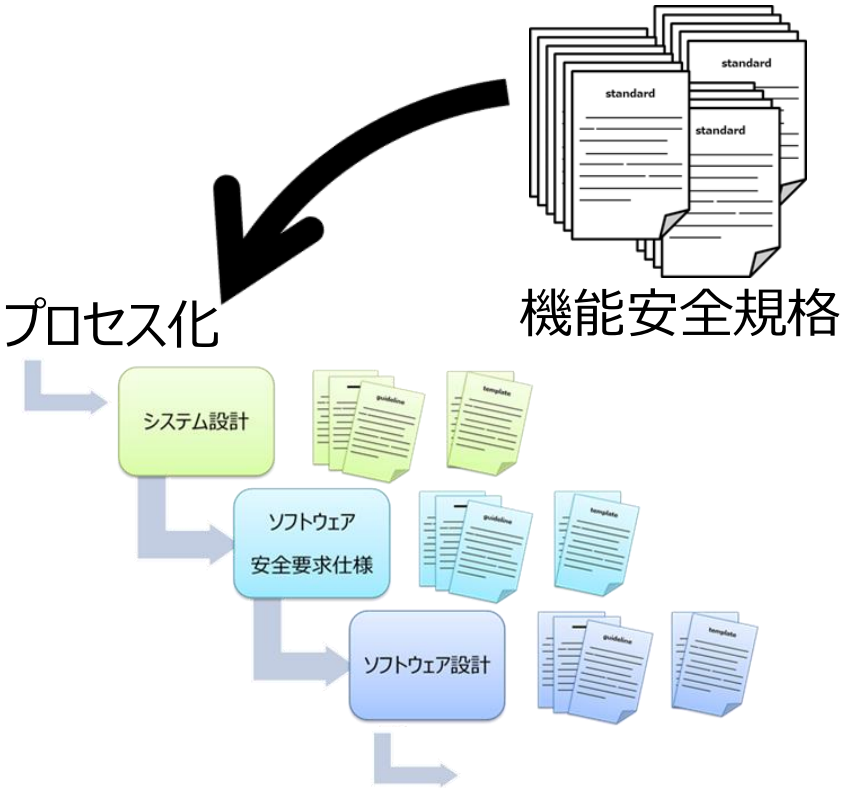
機能安全開発とは



リスクを受容できるレベルに

➢ その論理的説明が必要

機能安全対応へのアプローチ



機能安全プロセスに従った成果物

➢ 説明可能な状態を実現

背景～開発プロジェクト～

プロジェクト概要

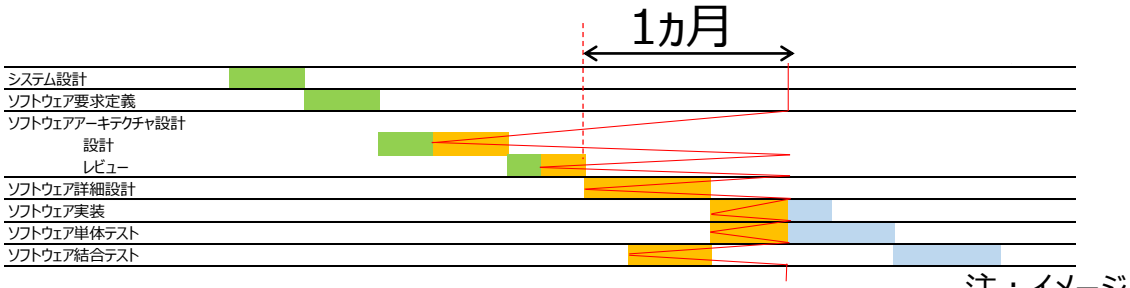
移動体ロボットの機能安全対応 ＋ 認証取得対応

- リーダー　機能安全開発3年
  - ソフトリーダー　機能安全開発1年以上、製品1種類
    - メンバー(3名)　一般製品開発 10年以上、機能安全講習受講済

ソフト担当を抜粋

開発状況

完了期限一ヵ月経っても、ソフトウェアアーキ設計書が完了しない



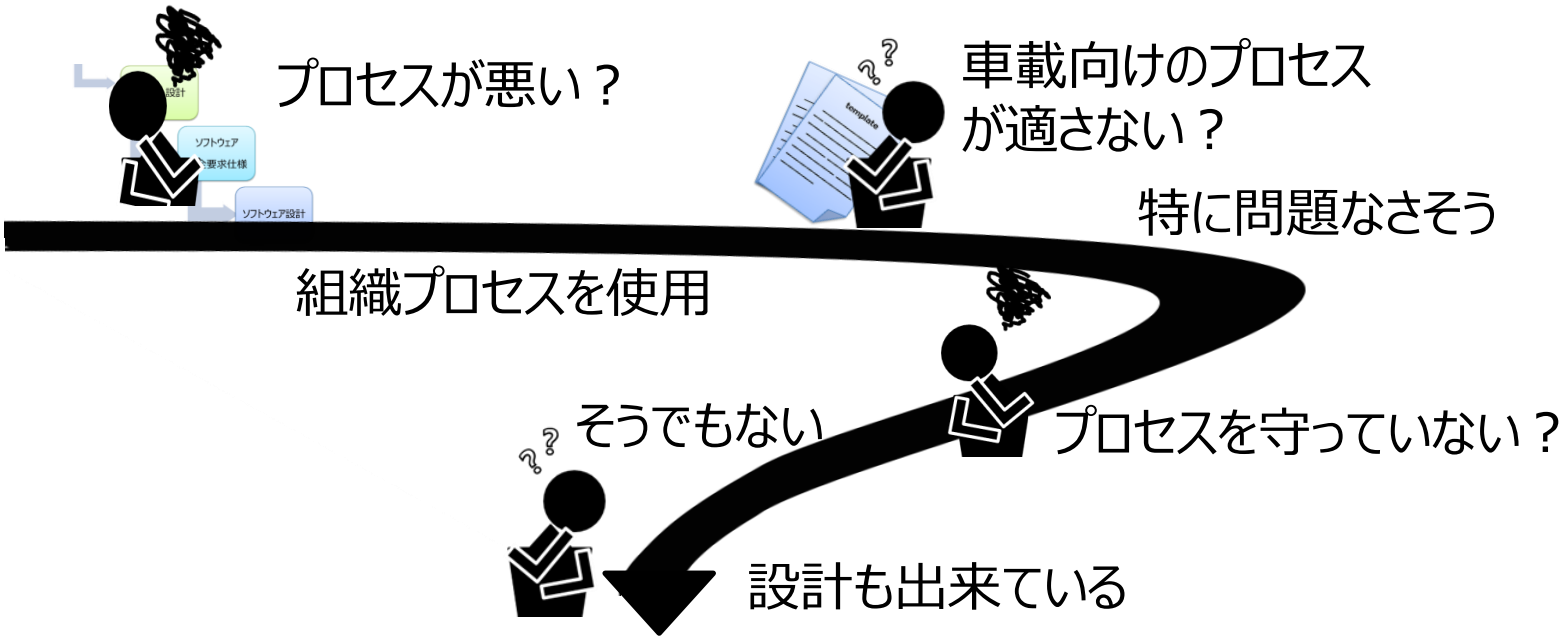
注：イメージ

プロジェクトの見直しが実施され、

私が現場に入り、改善することになった

何が悪いの？

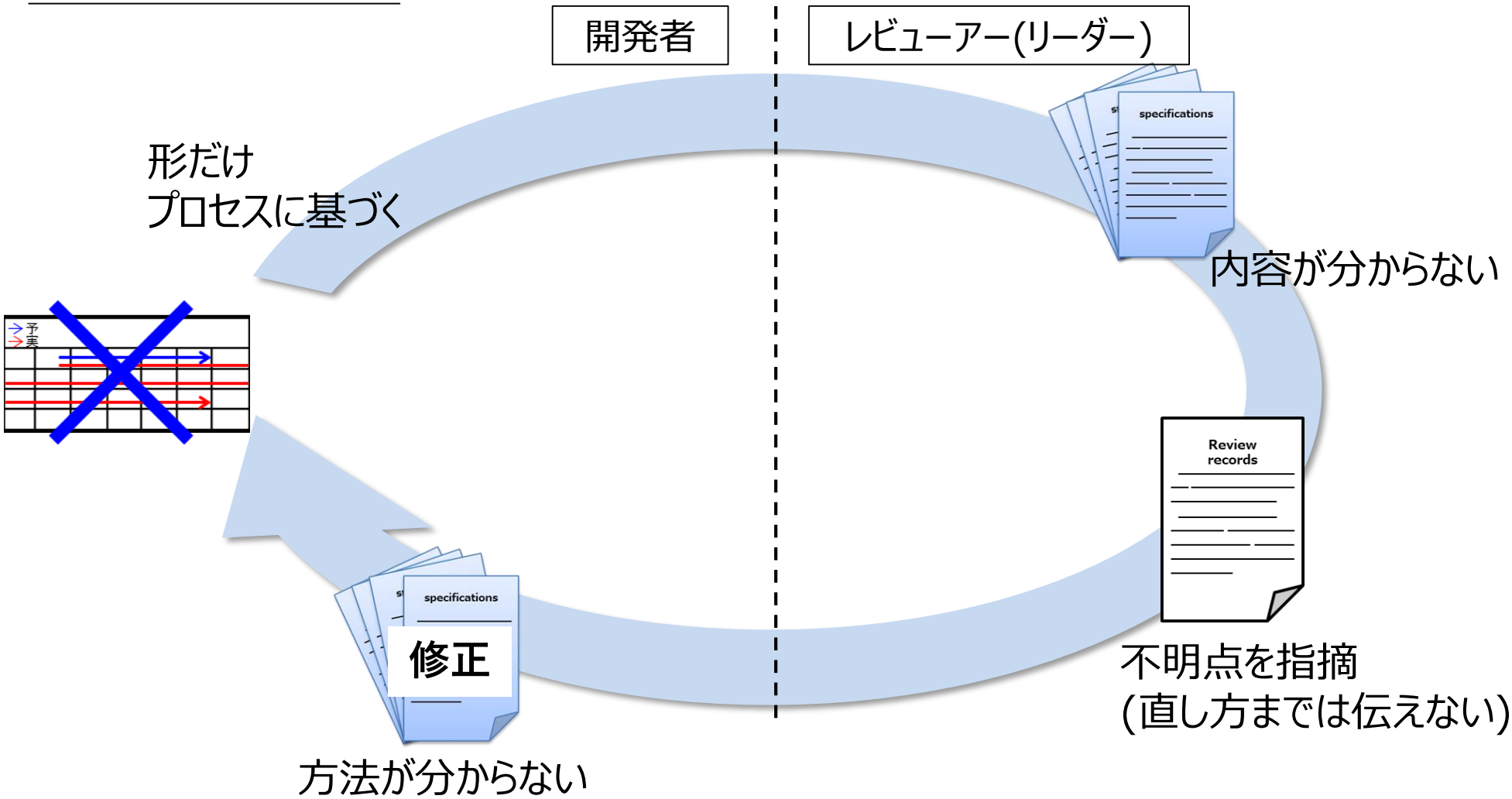
原因を探る



【仮説】 開発者の思い込みや、ガイドラインの解釈に問題があるようだ
→ 現物確認と、設計者／レビューアーのインタビューで分かった

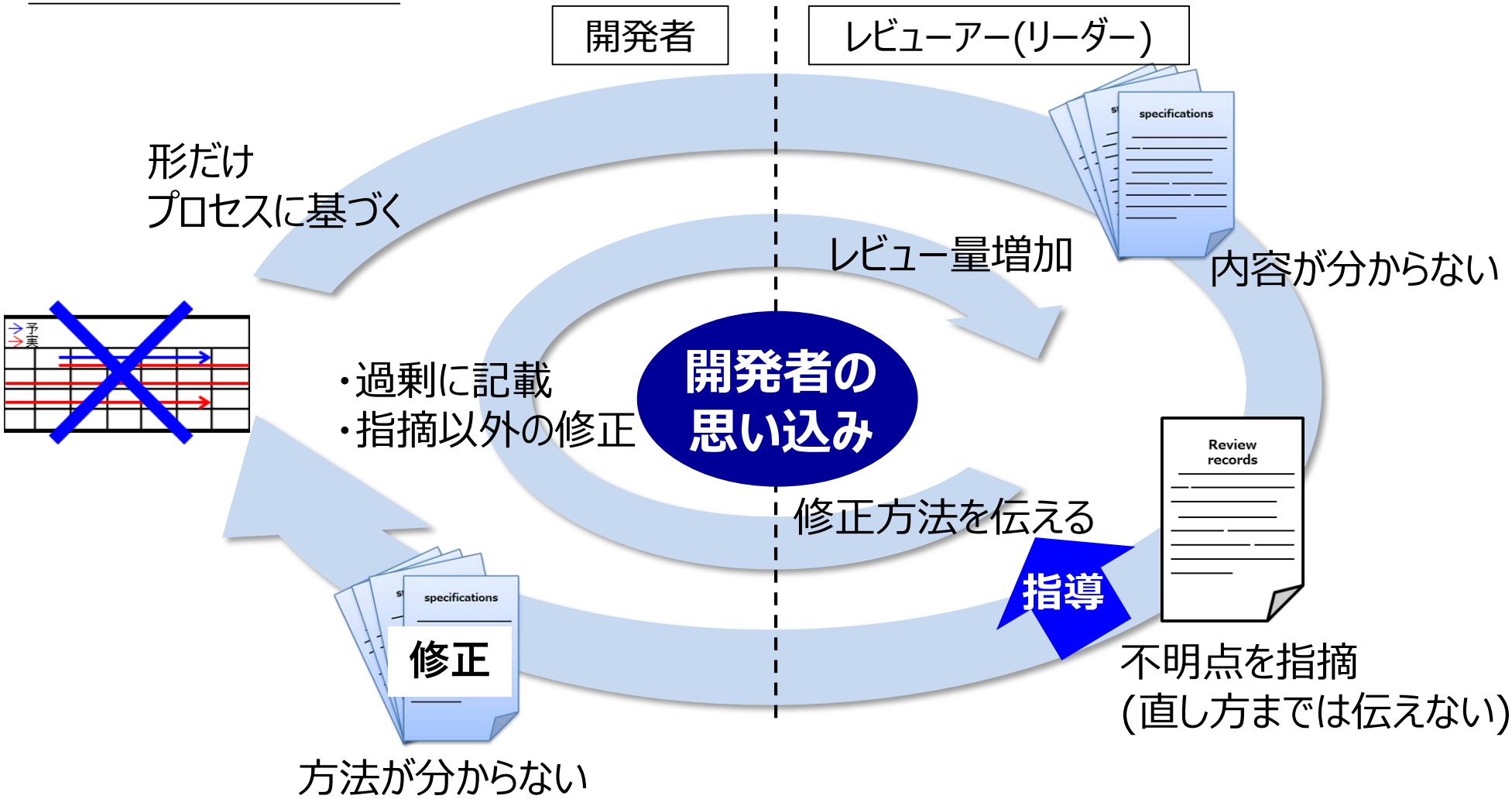
現場が陥っていた問題(支援前)

原因分析結果



現場が陥っていた問題(指導の結果)

原因分析結果



初期の指導の結果分かったこと

開発者(機能安全初心者)の意識	項目	レビューアー(機能安全経験者)の意識
自分が実装する時に苦労しそうなことを詳細に記載する	設計書	客観的に見て分かるように情報が網羅的に記載されている
プロセスを遵守してバグがないようにする	ゴール	成果物が理解でき正しいことが確認できるようにする

◆今回の取り組みポイント

- 設計書を利用する下流工程の開発者の視点でレビュー
- 「何が分からないのか」「どう修正したら良い」を具体的に伝えた

事例①「全体像より要素優先」(1/2)

改善前

サンプル：アーキテクチャ設計書の章立て

1. はじめに
2. システム構造
3. 設計方針
4. 静的構造
5. 動的構造
6. リソース見積り

・静的構造

- コンポーネント間
- コンポーネントA
- コンポーネントB
- ・・・

・動的構造

- コンポーネント間
- コンポーネントA
- コンポーネントB
- ・・・

大規模なソフトウェアなのにいきなり細分化されているので、適切かどうか判断できない

開発者

そうなった原因
　テンプレートの章立てを変更したらいけないと思っていた

事例①「全体像より要素優先」(2/2)

改善後

サンプル：アーキテクチャ設計書の章立て

1. はじめに
2. システム構造
3. 設計方針
4. 全体構造
5. コンポーネント詳細
6. リソース見積り

・全体構造(コンポーネントレベル)

- 静的構造
- 動的構造

・コンポーネント詳細

－コンポーネントA

- 静的構造
- 動的構造

－コンポーネントB

- 静的構造
- 動的構造

－・・・

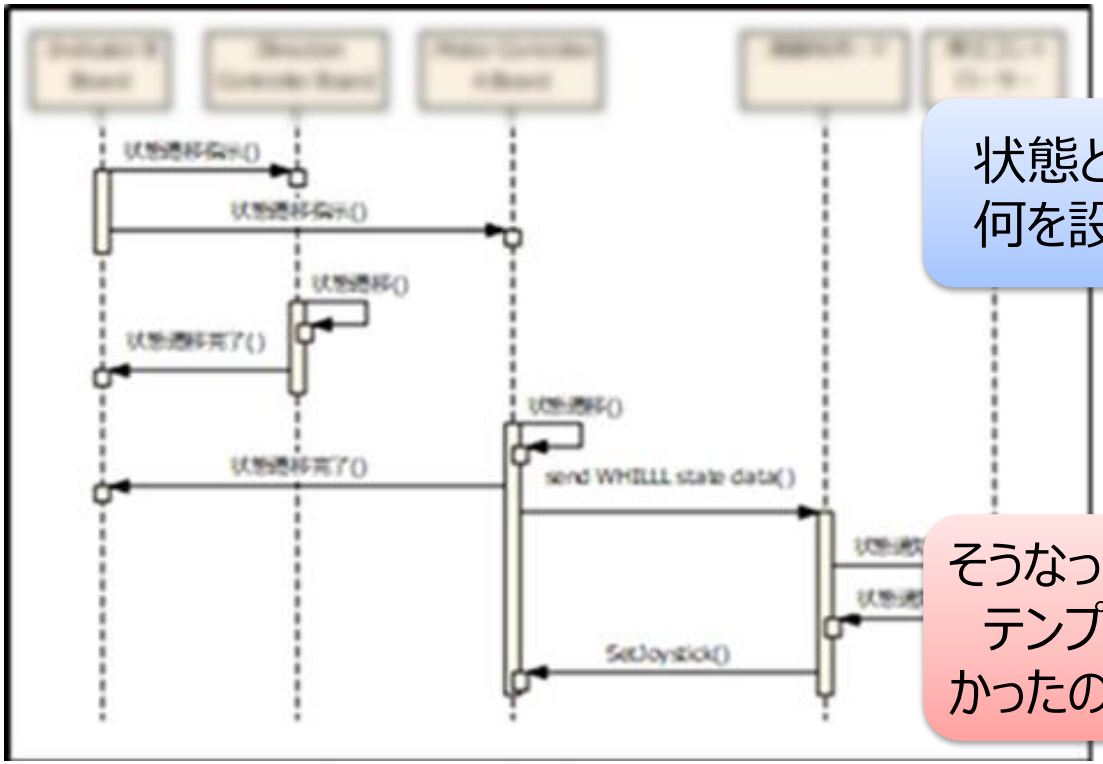
コンポーネントまでの設計と詳細を分けて記載するように指示した

機能の全体像が把握できた上で、詳細が理解できるようになった

事例②「記載例に追従」(1/2)

改善前

サンプル：動的設計



状態と処理を一度に表現しようとして、
何を設計しているのか分からなくなっている

開発者

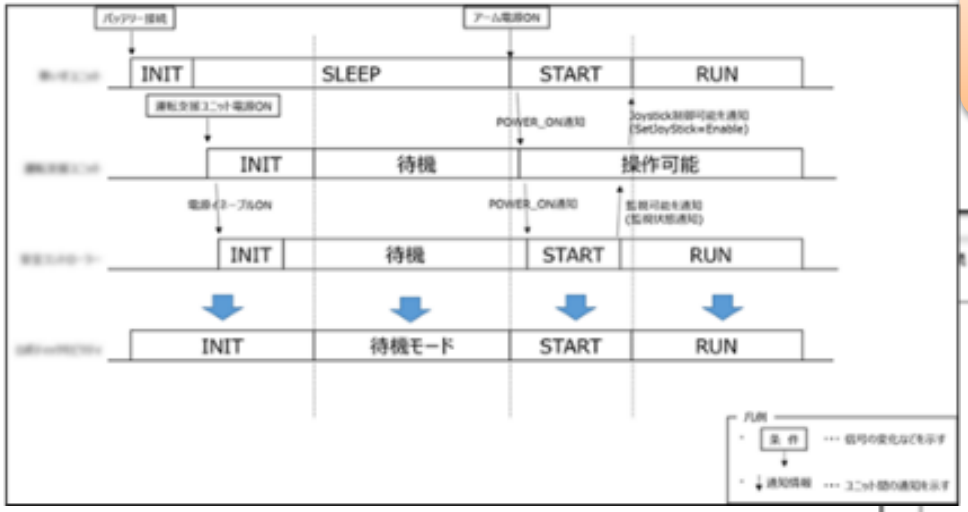
そうなった原因
　テンプレートにシーケンス図でしか記載がなかったので、その図を使うのが適切だと思った

事例②「記載例に追従」(2/2)

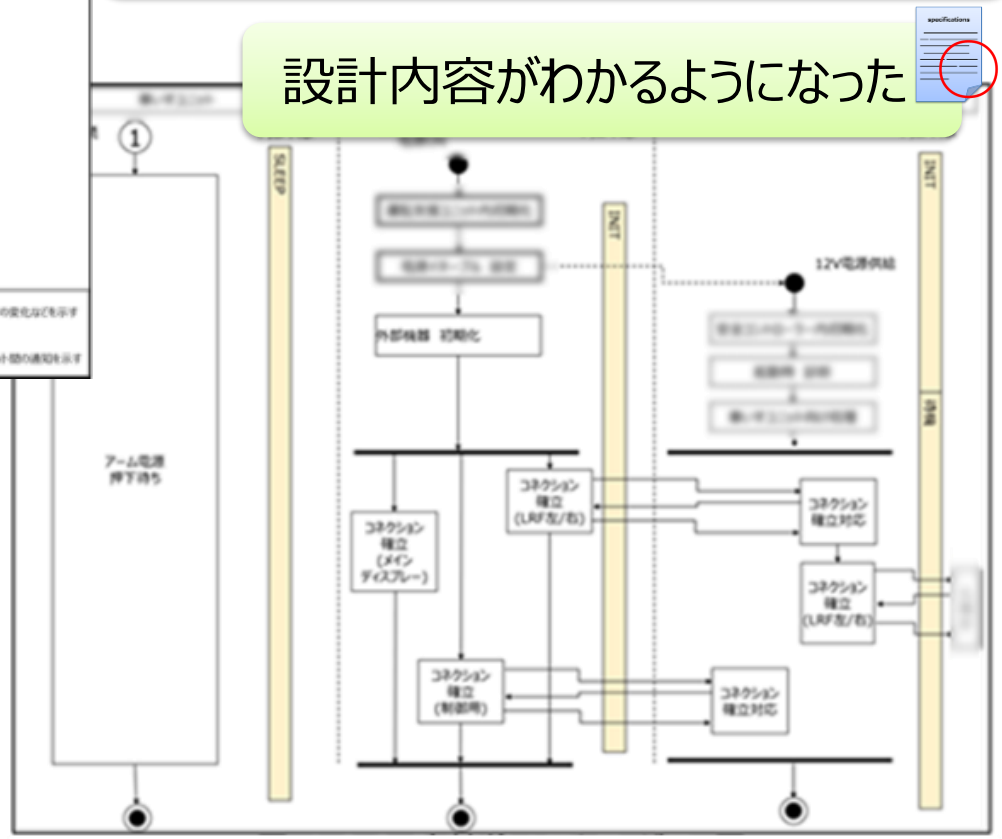
改善後

サンプル：動的設計

モード遷移と処理を分けて適切なモデル図で記載するように指示した



設計内容がわかるようになった



事例③「設計結果のみ記載」(1/2)

改善前

サンプル：タスク設計

表 5.7 タスク一覧

タスク名	周期	優先度	機能
1ms タスク	1ms	15	[illegible]
Ethernet 送受信タスク	event	5	[illegible]
10ms タスク	10ms	5	[illegible]
25ms タスク	25ms	3	[illegible]
50ms タスク	50ms	3	[illegible]
idle タスク	なし	0	[illegible]

動作判断に必要な条件が不足
- 処理時間
- 優先度の影響

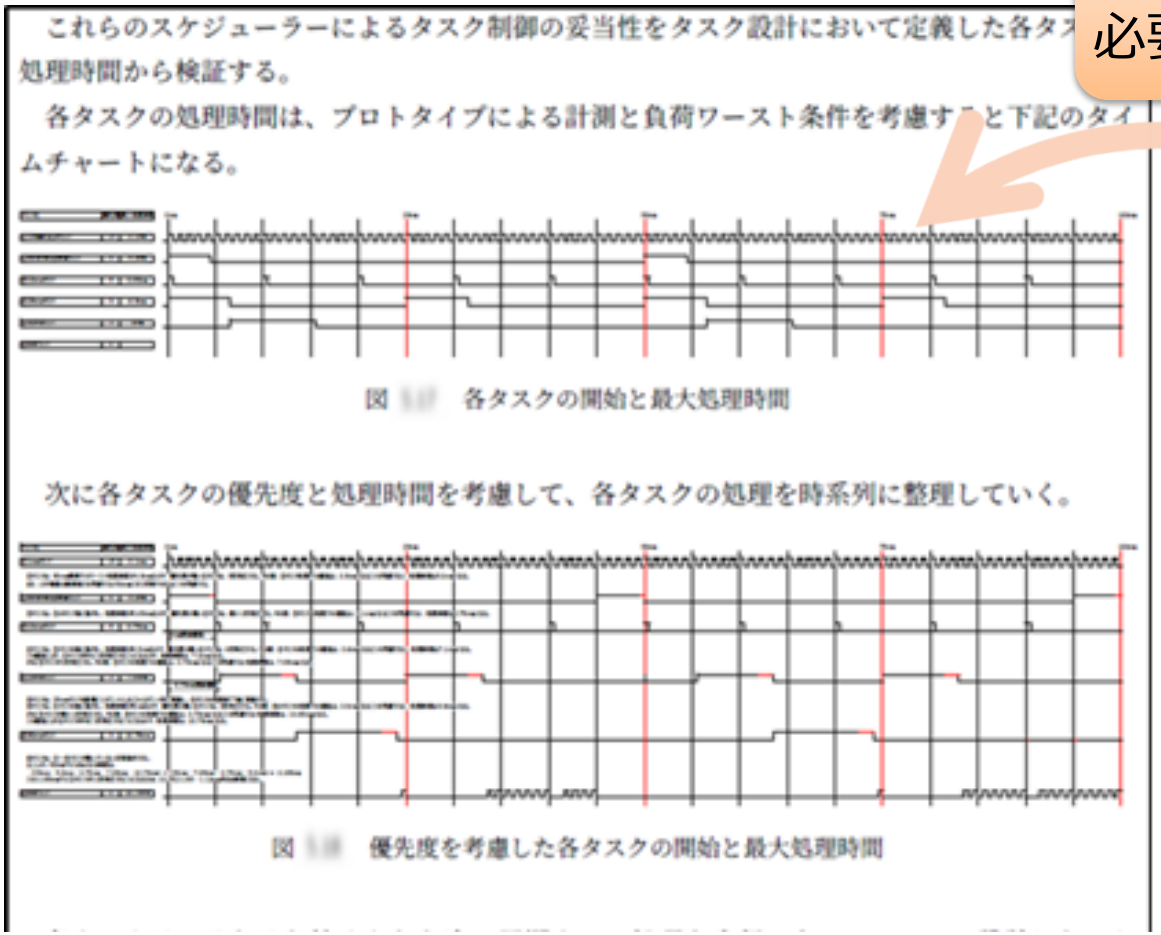
開発者

そうなった原因
　ガイドラインに記載されている内容は、記載したので、それで十分だと思った

事例③「設計結果のみ記載」(2/2)

改善後

サンプル：タスク設計



これらのスケジューラーによるタスク制御の妥当性をタスク設計において定義した各タス
処理時間から検証する。

各タスクの処理時間は、プロトタイプによる計測と負荷ワースト条件を考慮すると下記のタイ
ムチャートになる。

図 各タスクの開始と最大処理時間

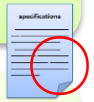
次に各タスクの優先度と処理時間を考慮して、各タスクの処理を時系列に整理していく。

図 優先度を考慮した各タスクの開始と最大処理時間

プロセスに記載されていなくても、
必要なものは記載することを説明した

タイミングチャート使用して
書くように指示した

動作判断に必要な情報が
記載された

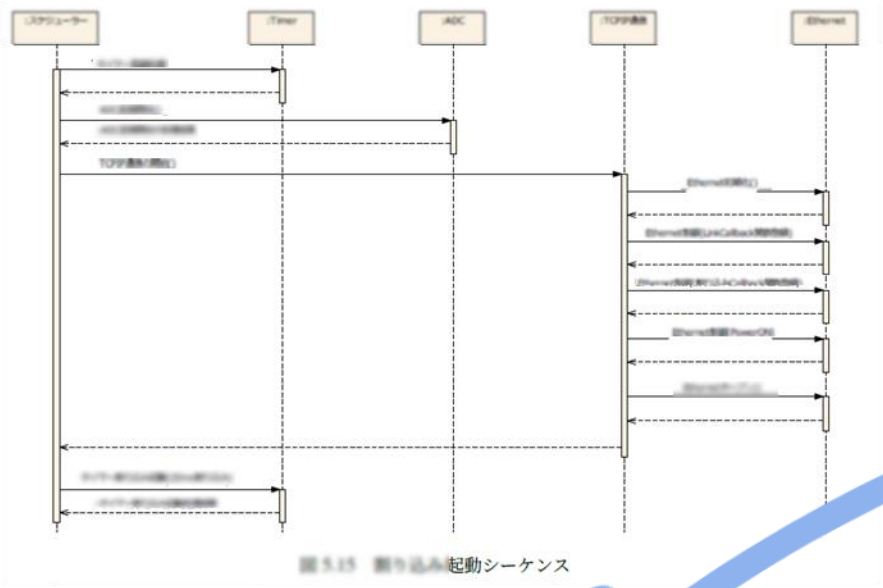


さらに
ビジュアル的に表現することで、
理解が容易になった

事例④「図に対する過剰な説明」(1/2)

改善前

サンプル：動的設計



図を補完する説明ではなく、
同じ内容の説明をしている

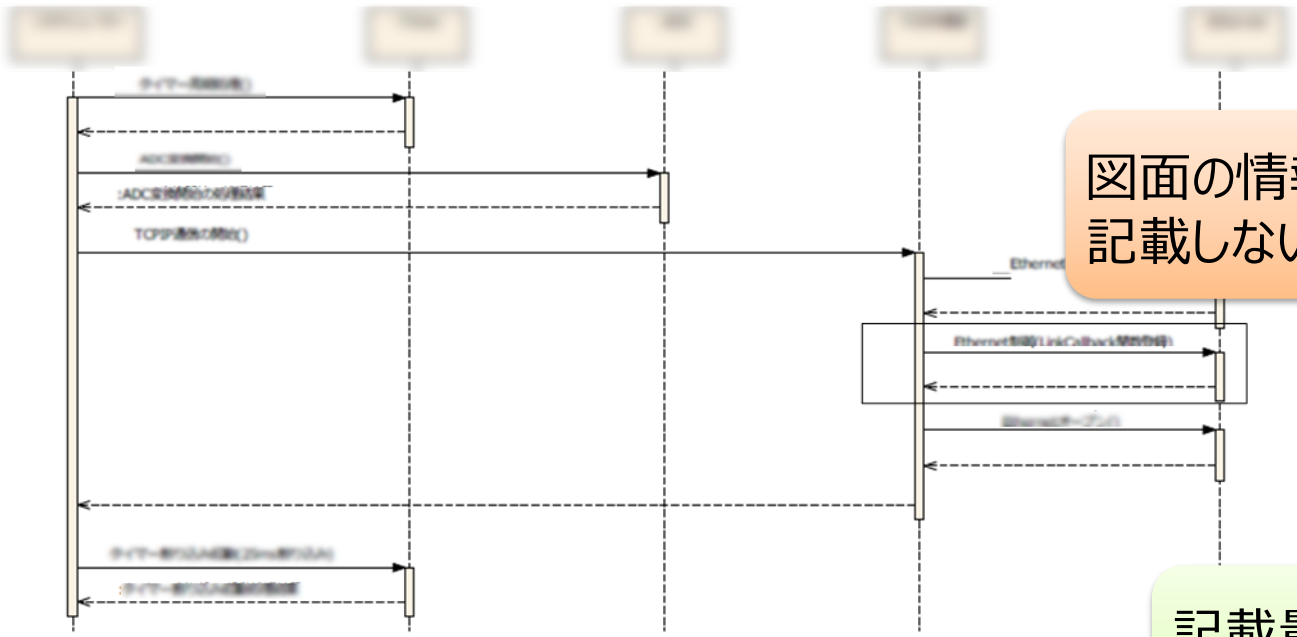
- 図と文章の不整合が発生し、レビュー量が増加

そうなった原因
　図だけでは、不十分だと思ったため、良かれと思って記載した

事例④「図に対する過剰な説明」(2/2)

改善後

サンプル：動的設計

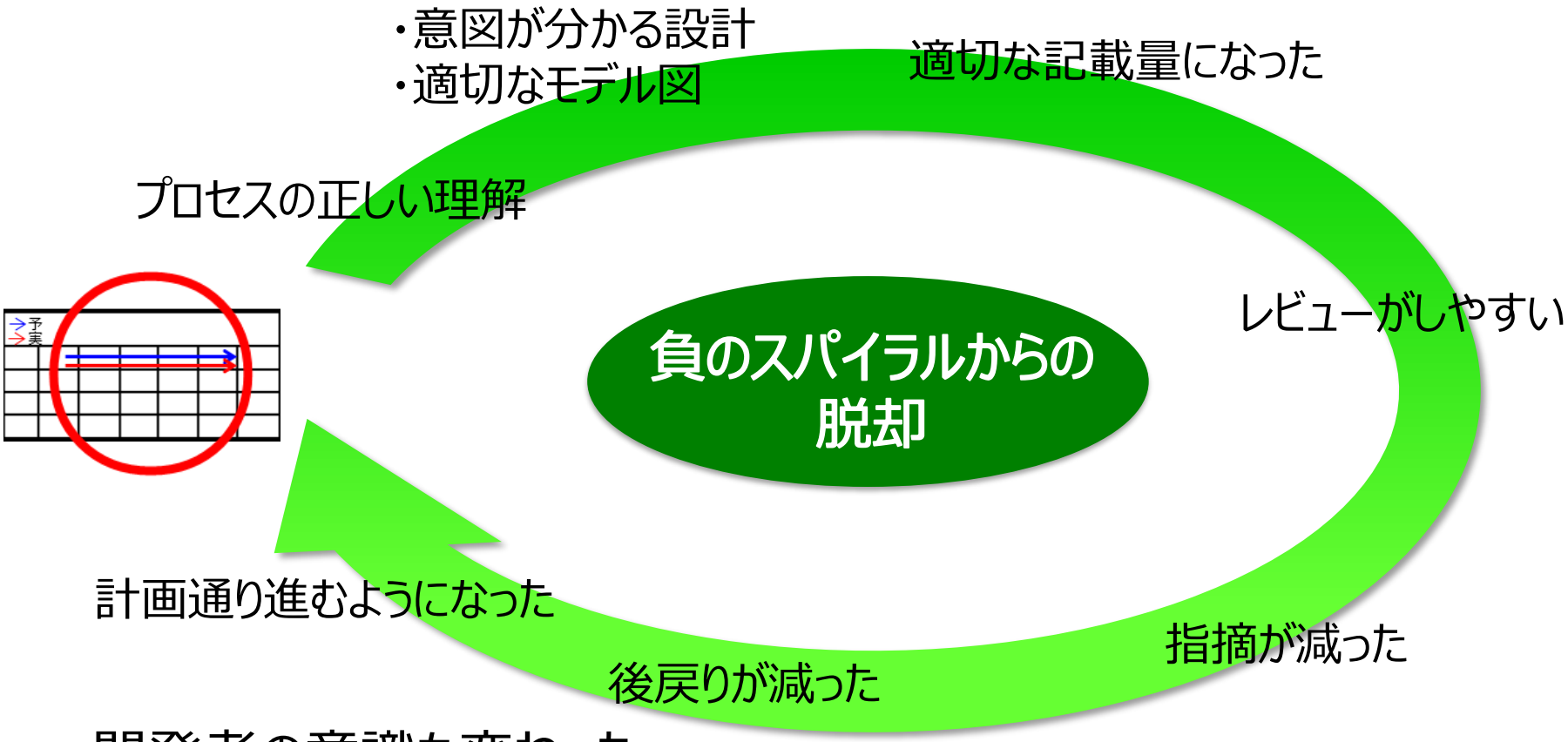


起動シーケンス

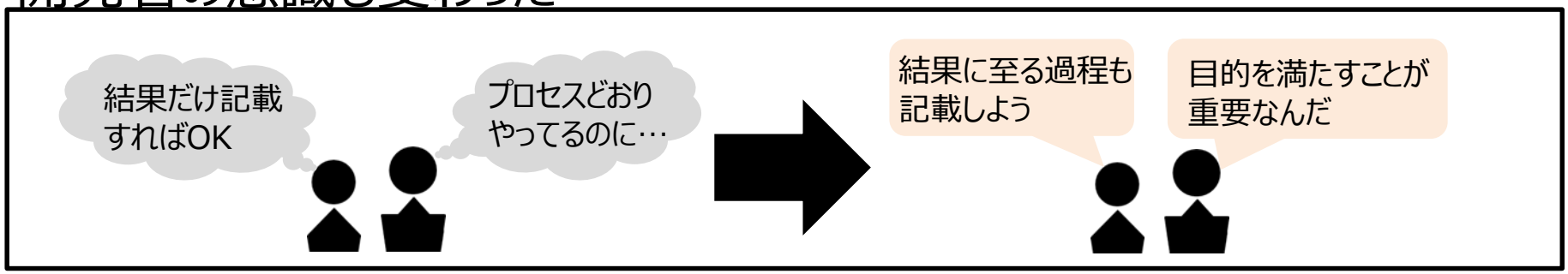
図面の情報を補完する内容以外は記載しないように指示した

記載量が減り、見通しが良くなりレビュー量も減った

改善の効果

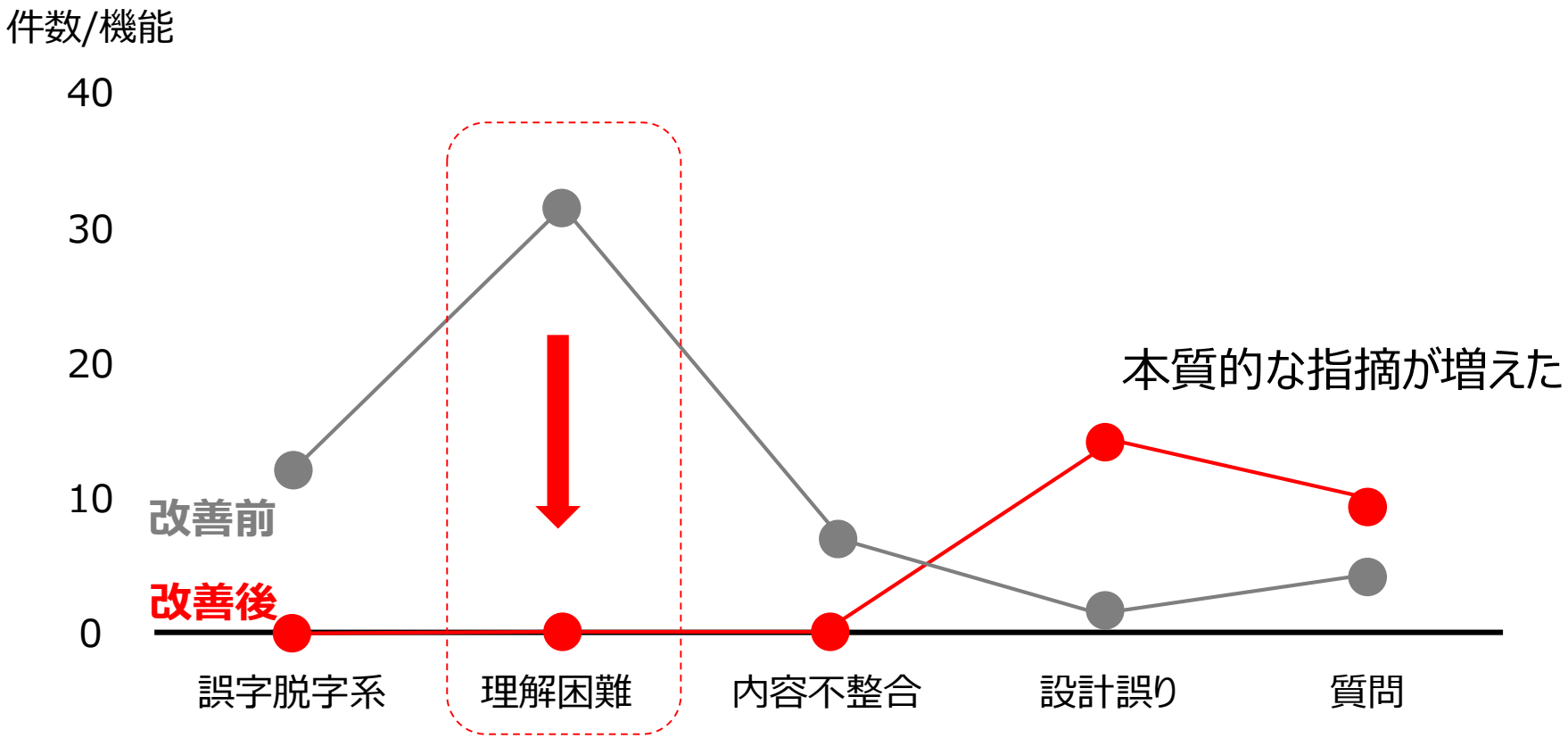


開発者の意識も変わった



改善の定量的効果

「説明できない設計書」から「説明できる設計書」へ



レビュー指摘件数と種別の比較

まとめ

機能安全開発で求められる説明できる設計ができるようになった

- ◆ 客観的に理解できるような情報が記載されている(理解容易性)
- ◆ 過剰な情報、重複する情報がなくシンプル(文書簡潔性)
- ◆ ビジュアルで分かりやすい(準形式的表記)

今回の取り組みで分かったこと

○プロセスが十分に整備されていても開発者はその通りにできるとは限らない

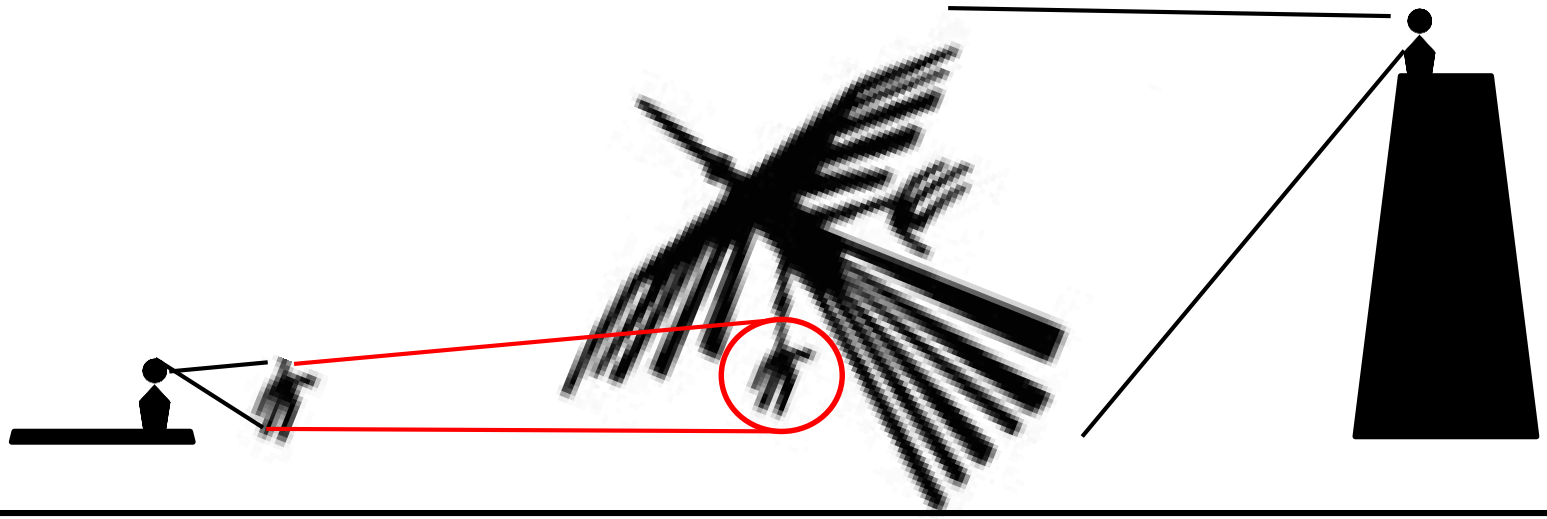
- ◆ 開発者はテンプレートやサンプルに引っ張られやすい
- ◆ プロセスを守っていても、実際は本質から外れている場合がある
  - ➢ 単なるトレーニングだけでは不十分、
    開発者が使えるようになるまでOJTを行う

○プロセスは継続的に改善する必要がある

- ◆ プロセス自体も開発者からのフィードバックに合わせて改善を行う
- ◆ SEPGは、現場を見て出来ているかを判断する

今後の抱負

開発者の視点とエキスパートの視点



多くの開発者	項目	エキスパート
結果を記載することが重要	意識	設計内容が伝えられることが重要
実装がイメージできないところから詳細化	設計	全体像から階層的に詳細化
与えられた手段だけで対応	道具	適切な手段を選択して対応

この考え方の違いを他のプロジェクトにも入り込み、
展開することで、開発者の設計力強化を実施する

ご清聴ありがとうございました