

# プロセス改善入門

—プロセス改善のためのツールの概要—

IPA 独立行政法人 情報処理推進機構  
SEC 技術本部ソフトウェア・エンジニアリング・センター

プロセス改善WorkingGroup NPT1/NPT6

委員 安達 賢二

- 
1. プロセス改善に活用できるツール
  2. なぜなぜ分析
  3. アセスメントモデル (SPEAK-IPA)
  4. SPINA<sup>3</sup>CH自律改善メソッド

# 1. プロセス改善に活用できるツール

# プロセス改善へのアプローチは多々あり

施策検討の糸口		打つべき施策	実施すべきこと
成功実績	有	自己組織で経験済みの施策	ベストプラクティスの導入・展開
	無	自己組織では未経験の施策	動向調査および先行評価
失敗経験	有	失敗経験に基づいた施策	プロセスの欠陥分析・原因分析
	無	アイデア提案型の施策	リスクを考慮
効果の定量化	有	達成状況が監視可能な施策	定期的に測定
	無	達成状況を監視可能にすべき施策	計数化を考慮
アセスメントモデル	有	アセスメント結果に基づいた施策	弱点の克服、長所の強化
	無	自己提案及びプロジェクト完了報告や反省会などからの施策	改善提案

# データ解析・表現に関する手法の事例

## ■QC7つ道具

品質管理分野で**数値データ**を整理・解析し、現象の定量的分析に役立つ手法

- ①特性要因図
- ②パレート図
- ③チェックシート
- ④ヒストグラム
- ⑤散布図
- ⑥管理図
- ⑦層別

## ■新QC7つ道具

主にスタッフ・管理者向け／**言語データ**を処理し、企画・計画立案に役立つ手法

- ①マトリクス図法
- ②系統図法
- ③連関図法
- ④親和図法
- ⑤PDPC法
- ⑥アローダイヤグラム
- ⑦マトリクス・データ解析法

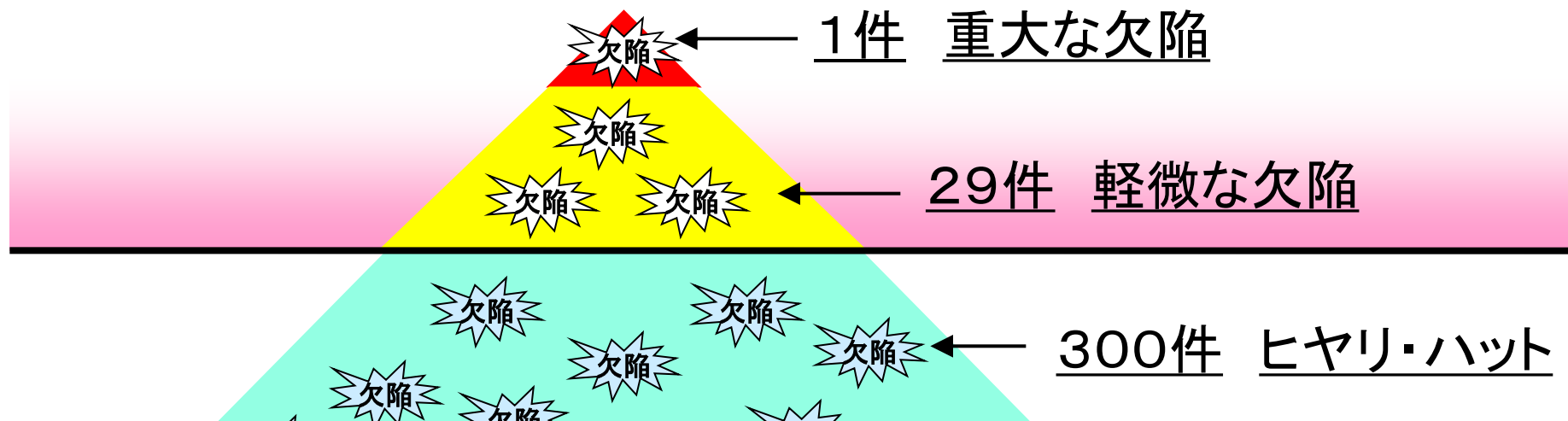
# プロセス改善に活用できるツール

	ベース 課題／モデル	対処課題 明確／不明確	アプローチ トップ／ ボトム
なぜなぜ分析	課題	明確	ボトム
SPINA <sup>3</sup> CH	課題＋ 部分モデル	明確	ボトム＋ トップ
アセスメント モデル	モデル	不明確	トップ

## 2. なぜなぜ分析

## ■ ハインリッヒの法則

- 1件の「重大な欠陥」の影には、29件の「軽微な欠陥」と300件の「インシデント(ヒヤリ・ハット)」がある

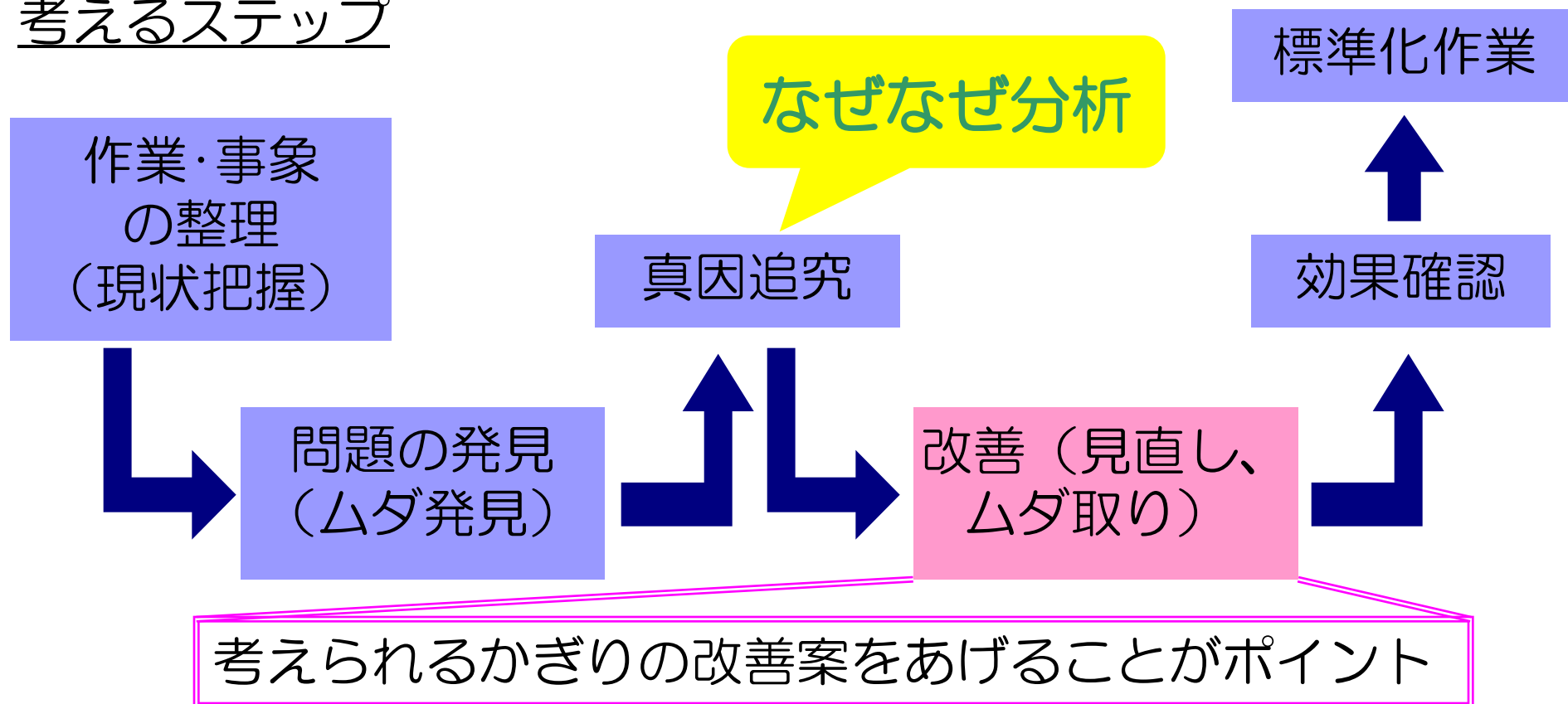


“重大な欠陥”はもちろんのこと、  
“軽微な欠陥”や“ヒヤリ・ハット”についても、  
真因を追究して改善するとよい



# 問題の真因を追求し最善の改善を導き出す

## 考えるステップ



考えられるかぎりの改善案をあげるには・・・

# 「なぜなぜ分析」とは・・・

- 現象を発生させている要因を思いつきで挙げるのではなく、もしなく出し切るために役立つ分析方法
- 真の要因に辿りつくまで「なぜ？」をトコトン繰り返し、最後の要因を裏返して再発防止策を立てる
- 「なぜ？」の深堀回数は決められていない



# 2つのアプローチ

---

## 1. あるべき姿からのアプローチ

- あるべき姿と問題になっている事象とを比較することでその問題を探っていく方向を決定後、「なぜ？」を繰り返しながら、その要因を探し出す。

## 2. 原理・原則からのアプローチ

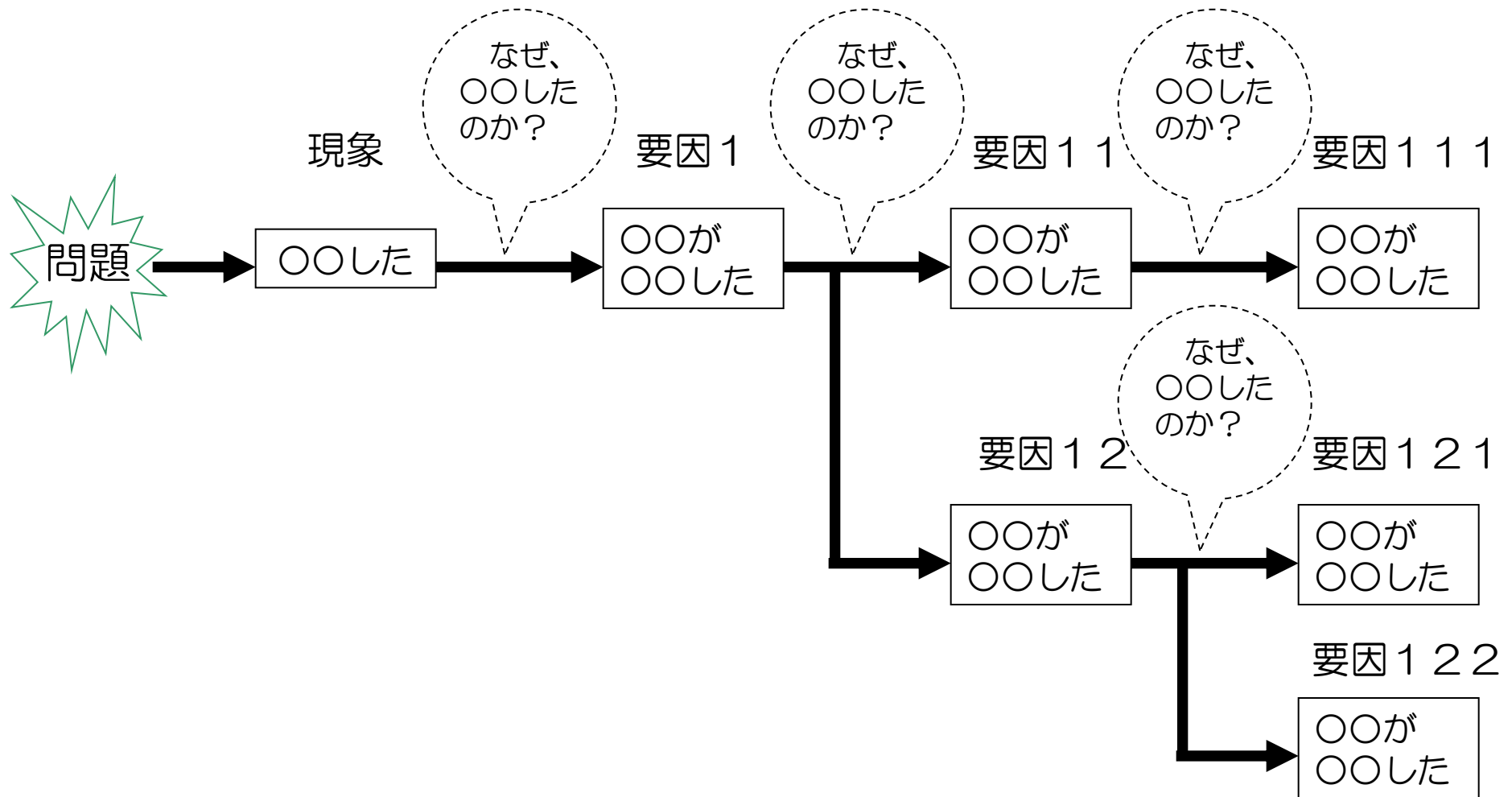
- 発生した問題の部分に焦点を絞り、そのトラブルを発生させる原理や原則といったものを「なぜ？」で掘り進むことで、より多くの要因を洗い出す。

- 原理 : 動作や仕組みを成立させる根本的な理屈
- 原則 : 原理を成立させるための条件

- 誰かを責めるような要因分析をしていないか
- 要因分析を始める前にゴールとなる真因を決めつけていないか
- 大半の現象は幾つかの要因が関係して起きているものだが、ひとつの要因導出で終わっていないか
- 要因分析は過去の過ち（経験）や自分の得意分野（知識・知見）に偏っていないか
- 複数の要因の中から費用対効果の高い対策を厳選しているか



1. 問題の抽出と事象の絞込み
2. 分析対象の背景や経緯を理解・把握
3. 範囲を限定するため前提条件の確認
4. 分析と検証（飛躍・矛盾がないか）の実施
5. 再発防止策の立案と評価
6. 再発防止策の実施と効果の確認
7. 類似問題の総点検と対策の横展開（標準化）



※イベントツリーのように要因展開するので、マインドマップを使うのも一案

## 1. 責任の所在は個人ではなく組織

- 再発防止策を導き出すために原因の追求をするのであって、当事者に対する責任の追及をしてはいけない

## 2. 参加者の選定

- 当事者はもちろんのこと、得意分野や経験が異なる人も参加するとよい
  - 現場のベテラン（管理者）
  - 仕組み・ルールの策定担当者
  - 品質保証の担当者
  - プロセス改善の担当者
  - 「なぜ<sup>2</sup>」に慣れたコーディネータ



## 3. 問題の整理

- 実行前に事実をしっかりと掴み、問題が発生した流れを整理し、仕組みや役割（機能）を理解する

## 4. 問題発生的前提となる条件の明確化

- 要因展開が無闇に発散するのを防ぐため、分析開始前に問題発生的前提条件（事実）を制約事項としてリストアップする



## 5. 主語をしっかりと書く

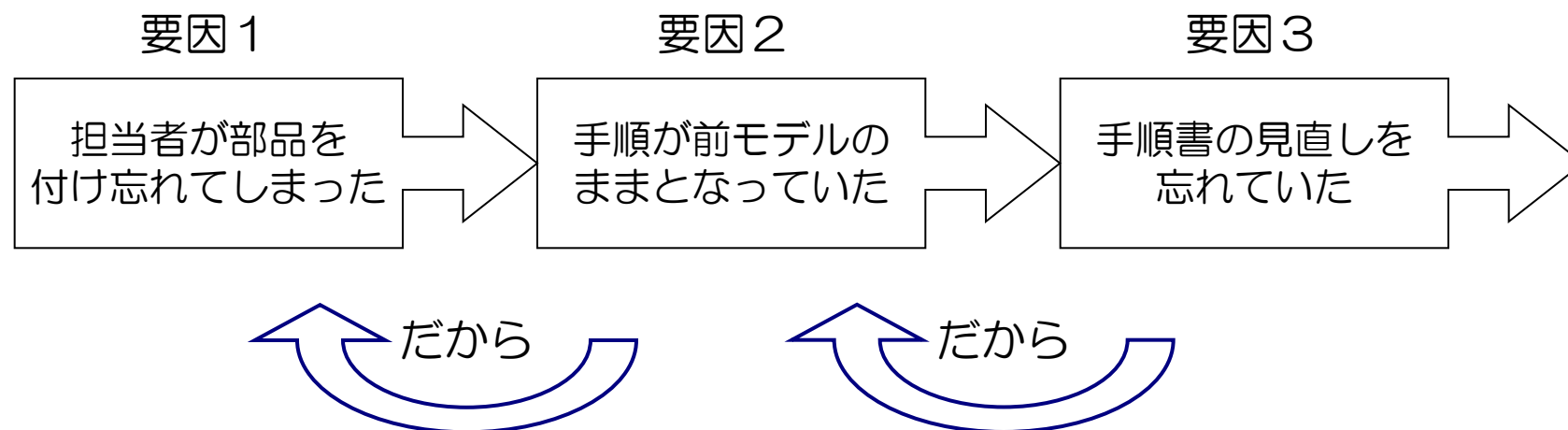
- 主語が欠けていると、次段の「なぜ？」を考える際に主体がぼやけるので、しっかりと明記する

## 6. 抽象的な表現は使わない

- 「～が悪い」という抽象的な言葉を使うと、人によって解釈（程度）が異なってしまいうため、明確かつ具体的な言葉に置き換えて表現する
  - 例：レビューのチェック項目が悪い
    - チェック項目の粒度が粗くて使えない？
    - チェック項目が何年も更新されず現状に合わない？

## 7. 要因分析の展開に飛躍はないか

- 分析終了後、最後の「なぜ？」から最初の「現象」までを逆向きに読んで違和感がないか確認する
  - ▶ 遡るときは「だから」という接続詞を用いて、  
(後段要因) だから (前段要因) というように読む



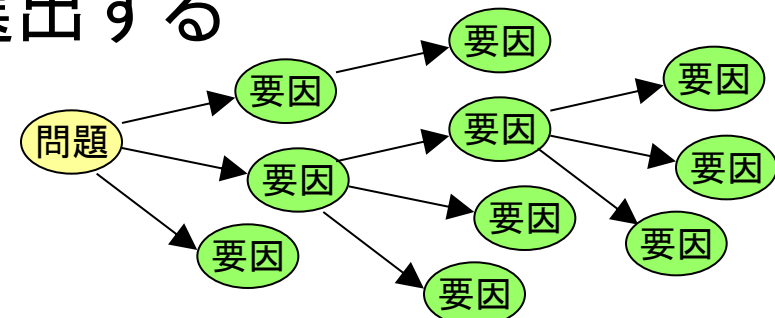
## 8. 心理面の要因で終わらない

- 人間の心理面の要因を最終要因とせず、「なぜ、〇〇したのか？」ともう一段掘り下げる
  - ボーッとしていた
  - 疲れていた
  - 気づかなかった
  - 思い込んでしまった、など

## 9. 要因の出し切り

- 導出要因が発生しなければ、前段の要因や現象が発生しないか、環境・プロセス・人・管理などの多面的な観点で要因が展開されているか確認する

1. 最近、身の回りで困った問題を一つ挙げる
2. 前提条件を挙げる
3. 問題をトップ事象として、イベントツリーの裾野が広がるようになぜ<sup>2</sup>を繰り返して要因分析を展開する
4. 後段要因を掘り下げたときに「後段要因だから前段要因である」と逆読みして違和感がないか確認する
5. 要因の掘り下げが出来なくなったら、要因を裏返して対策を導出する → 仕組み(プロセス)に落とし込む
6. 最も費用対効果の大きな対策を選出する



### 3. アセスメントモデル (SPEAK-IPA)

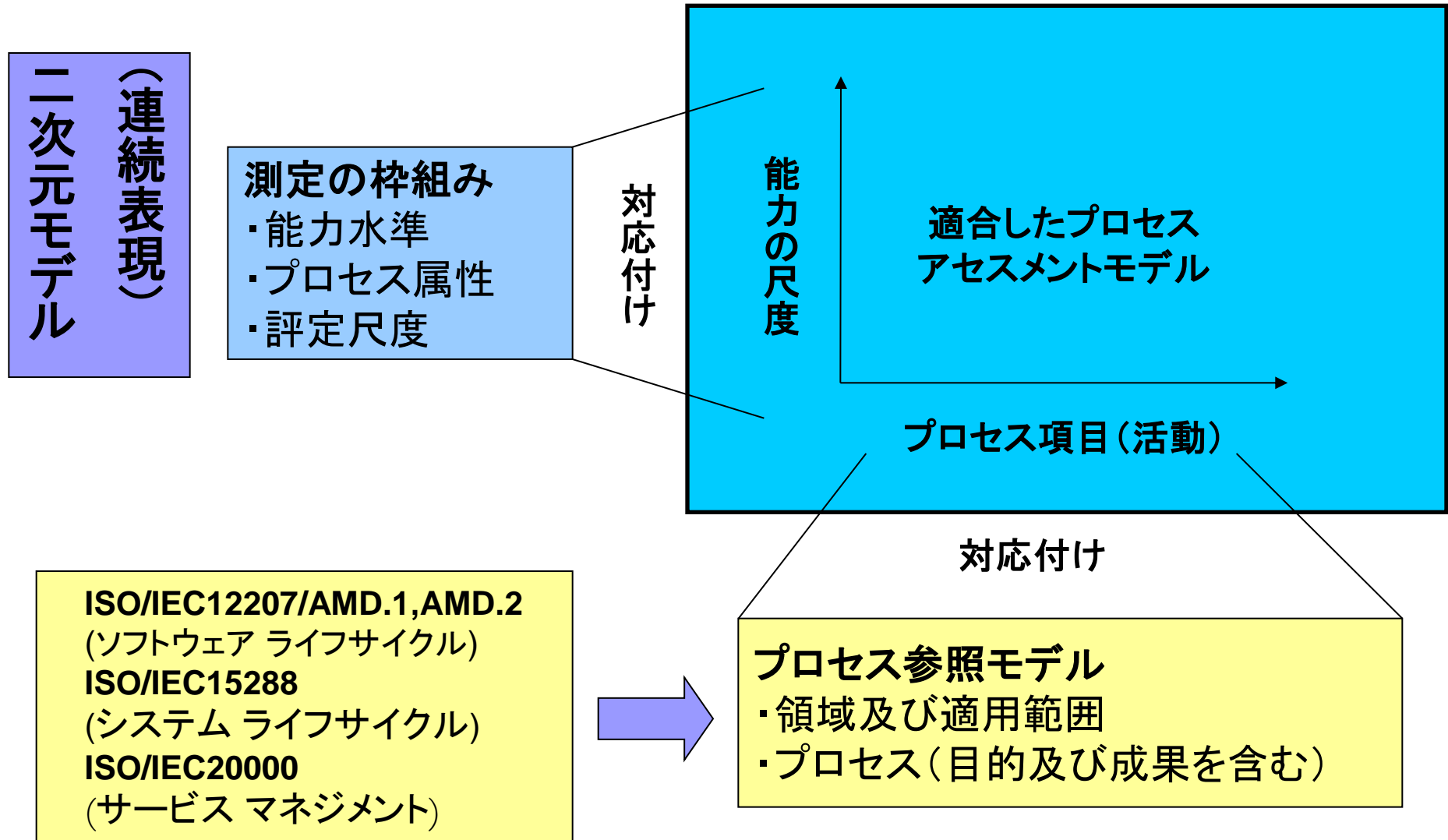
# いろいろなアセスメントモデル

---

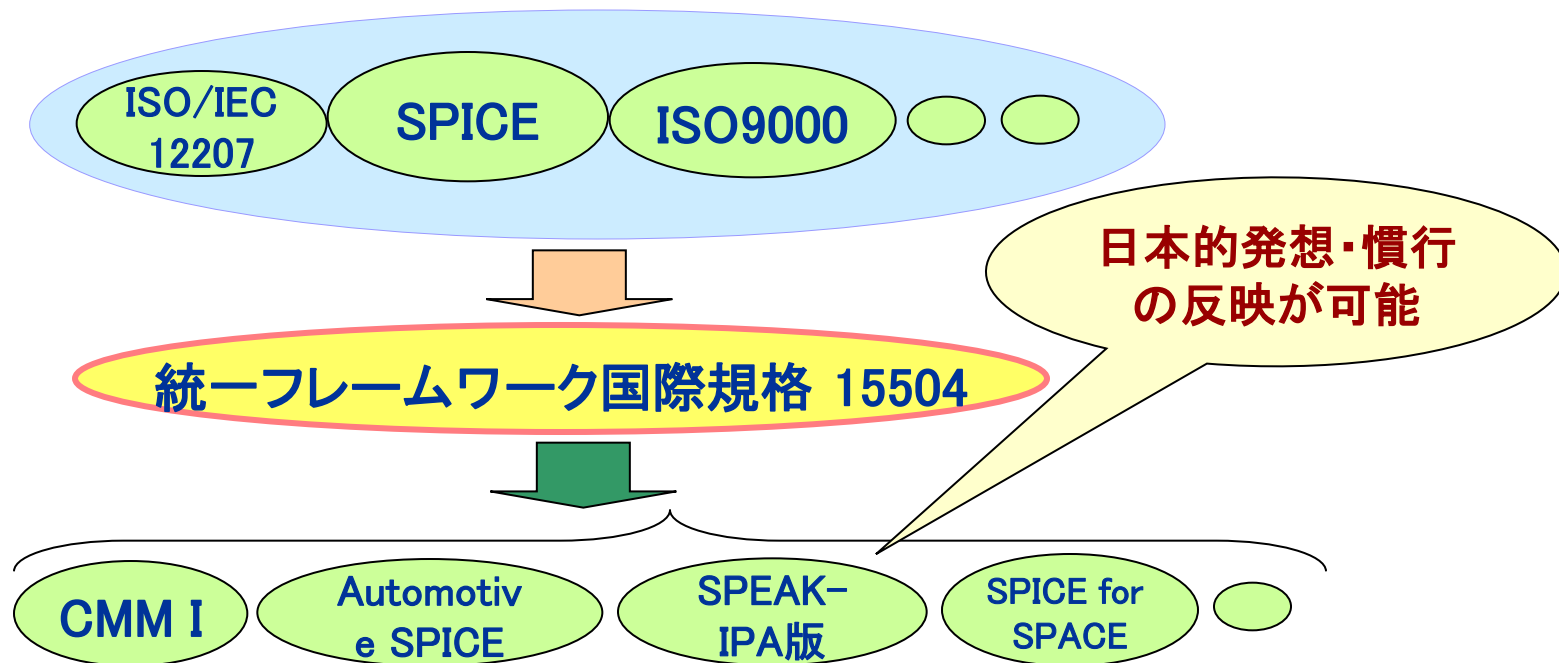
## ベストプラクティスに基づくプロセス改善 アセスメントによる改善機会の抽出

- CMMI®
- ISO/IEC 15504 (プロセスアセスメントの国際標準)
  - ISO/IEC 15504-5 (ソフトウェア)
  - ISO/IEC 15504-6 (システム)
  - ISO/IEC 15504-8 (サービスマネジメント)
- Automotive SPICE™ (自動車)
- SPICE for Space™ (宇宙機)

# ISO/IEC15504の構造



- ISO/IEC 15504は、プロセス改善と能力判定のためのアセスメント体系を規定する国際標準
  - ▶ プロセス能力を議論するための組織、国家を超えた共通基盤を提供
- SPEAK-IPA版は、ISO/IEC 15504の要件を備えたアセスメント方式の1つ





## ■ ISO/IEC 15504に準拠した日本発のモデル

### ➤ 開発の経緯

- 新日鉄ソリューションズ株式会社がSPEAKを開発(第1版:2002年3月)
- 社団法人情報サービス産業協会(JISA)がSPINACHを開発(2003年)
- 両者をベースに、IPA/SEC プロセス改善研究部会が一般化を行い、2007年9月に公開
- 実証実験に基づき改訂を行い、2011年3月に改訂版を公開

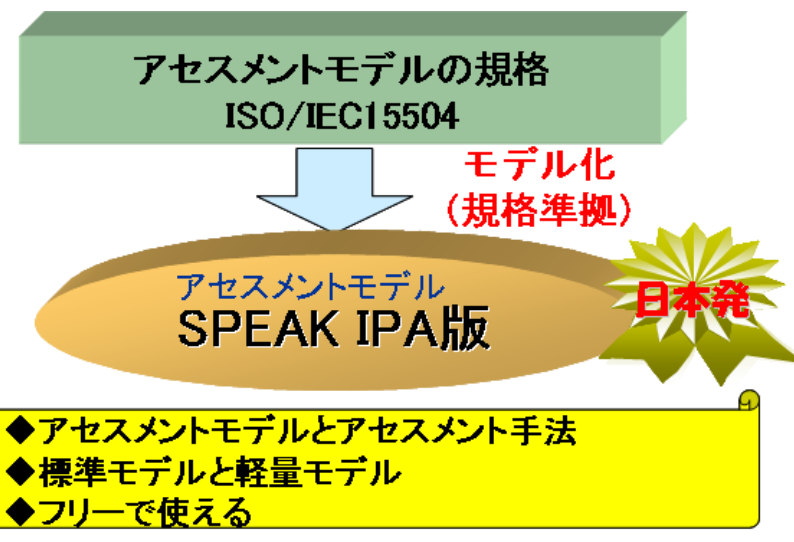
## ■ 多分野でISO/IEC 15504に沿ったアセスメントモデルを作成する場合の参考例として活用可

## ■ アセスメントの厳格さに応じた標準モデルと軽量モデルを提供

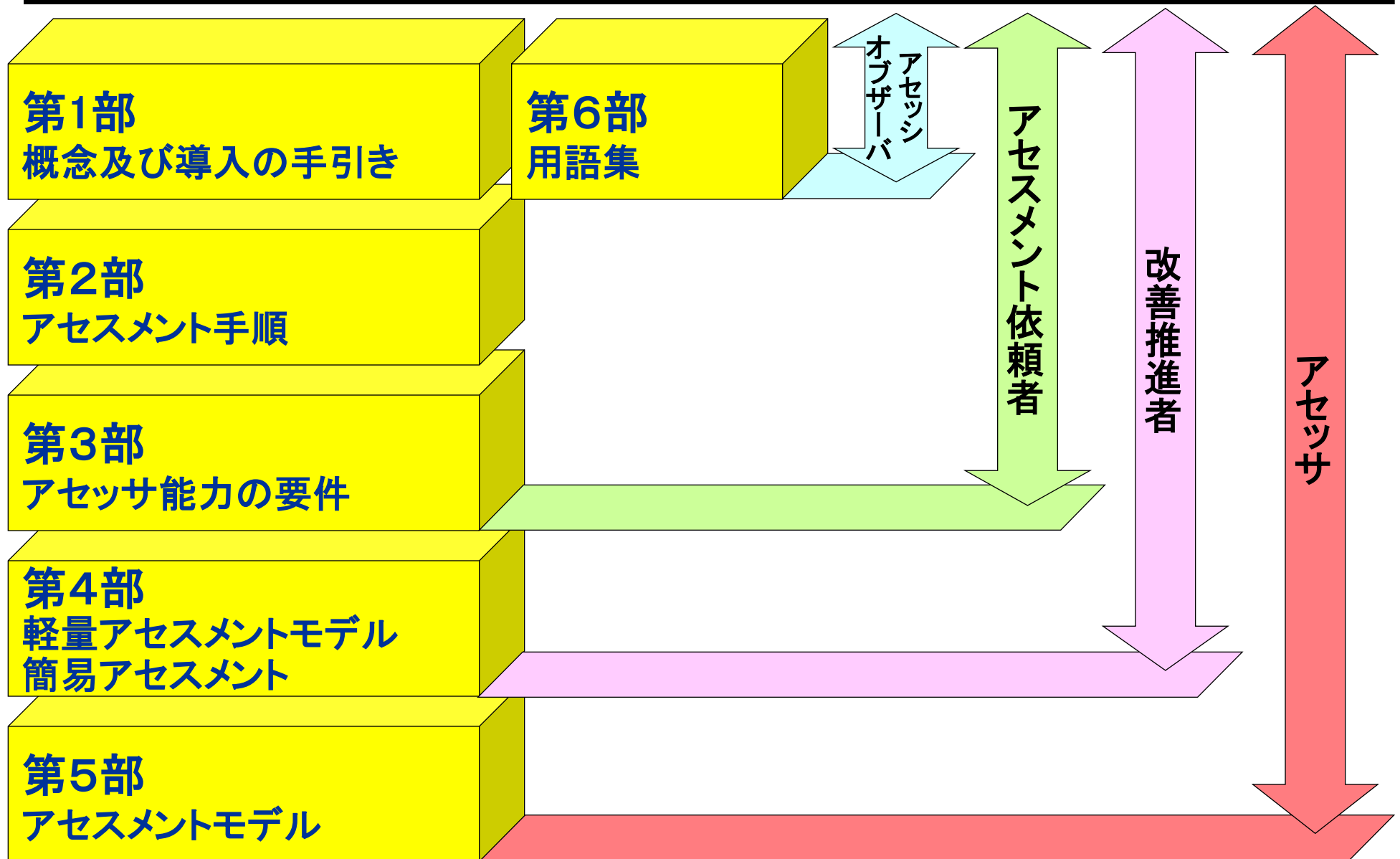
## ■ アセスメント手順を提供

## ■ フリーにダウンロード可能

<http://sec.ipa.go.jp/reports/20110328.html>



# SPEAK IPA版の体系



第1部: 概念および導入の手引き	ソフトウェアプロセスを診断すること、すなわちソフトウェアプロセスアセスメントの枠組みを提供している。この枠組みは、ソフトウェアの取得、供給、開発、運用、発展、および支援を計画、管理、監視、制御、および改善しようとする組織および/またはプロジェクトが利用することを想定している。
第2部 アセスメント手順書	第5部のアセスメントモデルあるいは第4部の軽量アセスメントモデルを利用して、プロセス改善あるいはプロセス能力判定を目的とした、プロジェクトあるいは組織としてのソフトウェアプロセスアセスメントを実施するための手順を定義したものである。
第3部: アセッサ能力の要件	SPEAK-IPA版に基づくソフトウェアプロセスアセスメントを実施する下記アセッサの能力に関する事項について定義したものである。 －適合アセッサ、リードアセッサ候補、リードアセッサ SPEAK-IPA版を利用した適合アセスメントを実施する際に、アセッサが適格性を有することを判断するときに用いる基準として、アセッサ能力に関する手引きとアセッサ認証制度についての試案を提供するものである。
第4部: 軽量アセスメントモデル・簡易アセスメント	主にプロセスアセスメントに関する基本教育を受けた人が自組織のプロセス能力診断につかうことを想定した軽量モデルとその診断手法を提供する。
第5部: アセスメントモデル	モデル要素対応表の説明および利用の手引き並びにモデル要素対応表からなる。SPEAK-IPA版のモデル要素対応表は、ISO/IEC 15504 適合のプロセスアセスメントモデルそのものであり、アセスメント(インタビュー)の際に参照するときに便利のようにプロセス参照モデルおよび測定 of 枠組みの目的および成果の原文を併記している。
第6部: 用語集	SPEAK-IPA版で使用する用語を定義している。

# プロセス項目（抜粋）

## 主ライフサイクルプロセスカテゴリ

- P.1.1 取得準備プロセス
- P.1.2 供給者選択プロセス
- P.1.3 供給者監視プロセス
- P.1.4 顧客の受入プロセス

### P.2 供給プロセス

- P.3.1 要求事項抽出プロセス
- P.3.2 システム要求分析プロセス
- P.3.3 システムアーキテクチャ設計プロセス
- P.3.4 ソフトウェア要求分析プロセス
- P.3.5 ソフトウェア設計プロセス
- P.3.6 ソフトウェア構築プロセス
- P.3.7 ソフトウェア結合プロセス
- P.3.8 ソフトウェアテストプロセス
- P.3.9 システム結合プロセス
- P.3.10 システムテストプロセス

### P.5 保守プロセス

## 支援ライフサイクルプロセスカテゴリ

- S.1 文書化プロセス
- S.2 構成管理プロセス
- S.3 品質保証プロセス
- S.4 検証プロセス
- S.5 妥当性確認プロセス
- S.8 問題解決プロセス

## 組織ライフサイクルプロセスカテゴリ

- O.1.1 組織に関するアライメントプロセス
- O.1.2 組織管理プロセス
- O.1.3 プロジェクト管理プロセス
- O.1.4 品質管理プロセス
- O.1.5 リスク管理プロセス
- O.1.6 測定プロセス
- O.4.1 人的資源管理プロセス
- O.4.2 教育訓練プロセス
- O.7 ドメイン技術プロセス

ところで、

アセスメントモデルを使用して、現状のよしあしを評価するアセスメントは、監査と同じですか？

⇒ **似て非なるもの**



監査は、定められた遵守すべきルールや規程に対して、実際の業務やその成果物がそれらに則っているかどうかを判定すること。

監査における、ルールや規程＝“要求事項”  
逸脱すれば是正する！

アセスメントは、ある標準(モデル)に対して、対象とするソフトウェア開発組織の<強み>や<弱み>を分析し、改善の機会や改善のリスクを確認すること。

アセスメントにおける、モデル≠“要求事項”  
モデル通りに実施されていることが正じゃない。  
結果を、どのように使って改善するのかわからないのかは、  
アセスメントを受けた人たち次第！

## ■ 強み

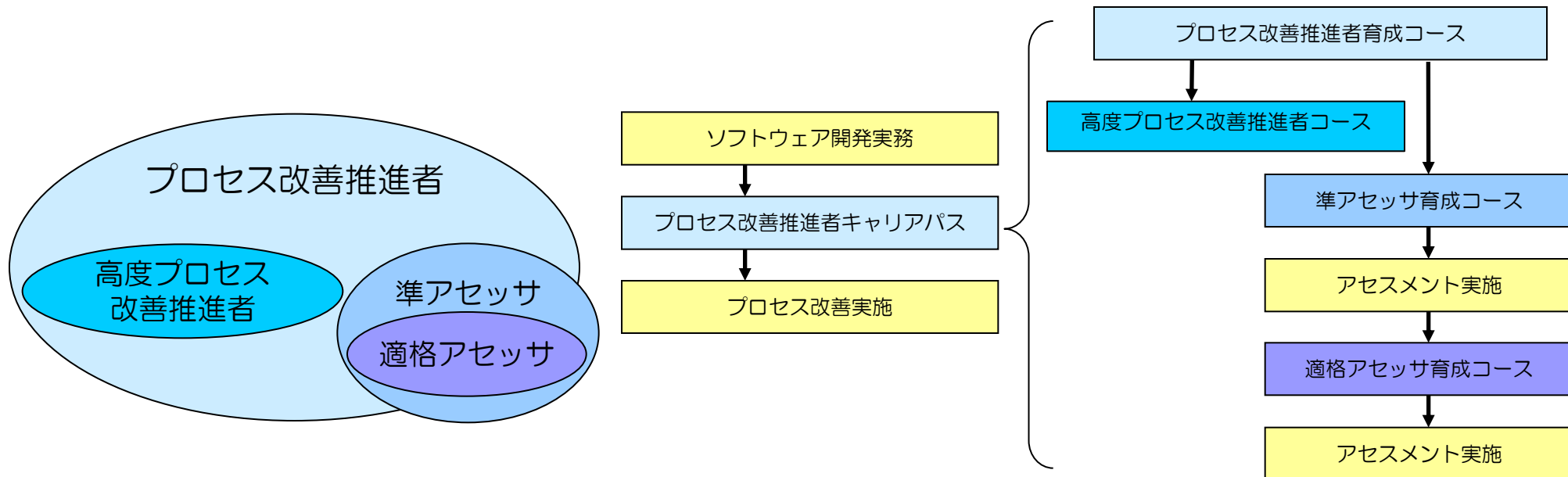
- 共通用語を確立する
- 共有されたビジョンを推進する
- ベストプラクティスを基に作成する
- アセスメントおよび改善のための枠組みを提供する
- ベンチマーキングを可能にする

## ■ 弱み

- 現実の世界を単純化している
- 完全に包括的ではない
- 使用に際して、判断および洞察が必要
- 事業のゴールに対してテーラリングしなければならない

# プロセス改善推進者のキャリアパス

- ◆ プロセス改善推進者育成コース
  - プロセス改善活動を牽引する推進者を育成する
- ◆ 準アセッサ育成コース（ベーシック）2日間
- ◆ 準アセッサ育成コース（アドバンスト）3日間
  - アセスメントモデルを適用した適合アセスメントにチームメンバーとして参加できるアセッサを育成する

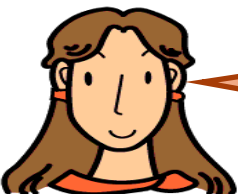


## 4. SPINA<sup>3</sup>CH自律改善メソッド



- ハードルの低いアセスメントモデル（初代SPINACH）

- 質問票構想（過去のSEIや日本の各社の例を参考に）



実際に作業しながら考えよう！！

モデルや他社経験も捨てたものじゃない！！

**ワークシート**を使ったプロセス改善のナビゲーション

# SPINA<sup>3</sup>CH自律改善メソッドとは

---

SPINA<sup>3</sup>CH = **S**PI with **N**avigation, **A**wareness,  
**A**nalysis, **A**utonomy for **C**hallenge

Navigation: ガイドがある

Awareness: 気づきから始める

Analysis: 状況・課題を分析して進める

Autonomy: 自律的に改善を進める

Challenge: 業務上のチャレンジ意識が重要

# SPINA<sup>3</sup>CH自律改善メソッドとは

ソフトウェア開発の「仕事の進め方」を現実に即して改善・向上させるためのツール

## 利用の場面

- ・ 開発チームの取り組みとして
- ・ エンジニア個人の自己トレーニング、自己チェックとして
- ・ 企業総体の改善ツールの一つとして

特色： 自主的な努力と追求（手と頭を動かす作業）を期待

しかし、その手がかりとしてのヒントも提供

**SPINA<sup>3</sup>CH**自律改善メソッドでは、次のような作業の仕組み・道具を用いる

- 開発の現場の「事実」と問題への「気づき」
- 注目するポイントを見出す
- 現実のプロセスの良いところと悪いところを考える
- 問題の原因や改善点の把握
- 良い事例や他の組織との比較
- ソフトウェア開発技法、マネジメント技法の学習
- ツールなどの調査・評価
- よく考え、その結果を改善計画としてまとめる
- 知識の蓄積と整理を行う

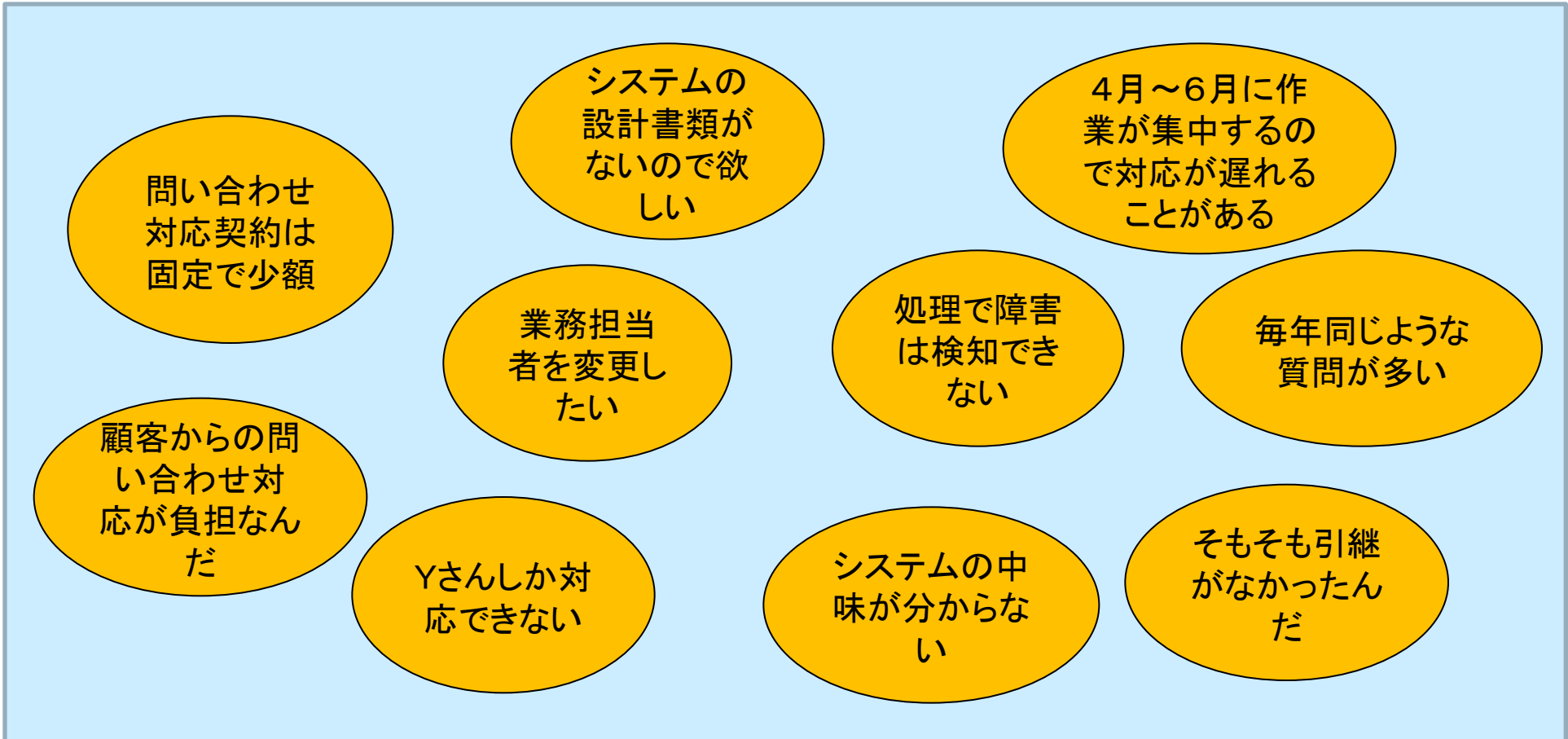
# SPINA<sup>3</sup>CH自律改善メソッドの特色(1/4)

## モデルベース改善と課題ベース改善の融合

	課題ベース	モデルベース
陥りやすい問題	<ul style="list-style-type: none"> <li>● 火がついた課題に着目するあまり、やるべき事に網羅性がない</li> <li>● 将来の課題につながる手当ができにくいいため、常に火消しの状態</li> </ul>	<ul style="list-style-type: none"> <li>● 一度に多量のプラクティスを示されるため、現場に敬遠される</li> <li>● レベル到達ありきになってしまう</li> <li>● モデルを実装してしまいがちで、現場には負担になりやすい</li> <li>● 推進側(SEPG)が進めてしまい、現場不在に陥りやすい</li> </ul>
メリット	<ul style="list-style-type: none"> <li>● 改善の初心者も入りやすい</li> <li>● 改善の効果が即体感できる</li> </ul>	<ul style="list-style-type: none"> <li>● ソフトウェア開発としてやるべきこととして網羅的にしくみの定義ができる</li> </ul>

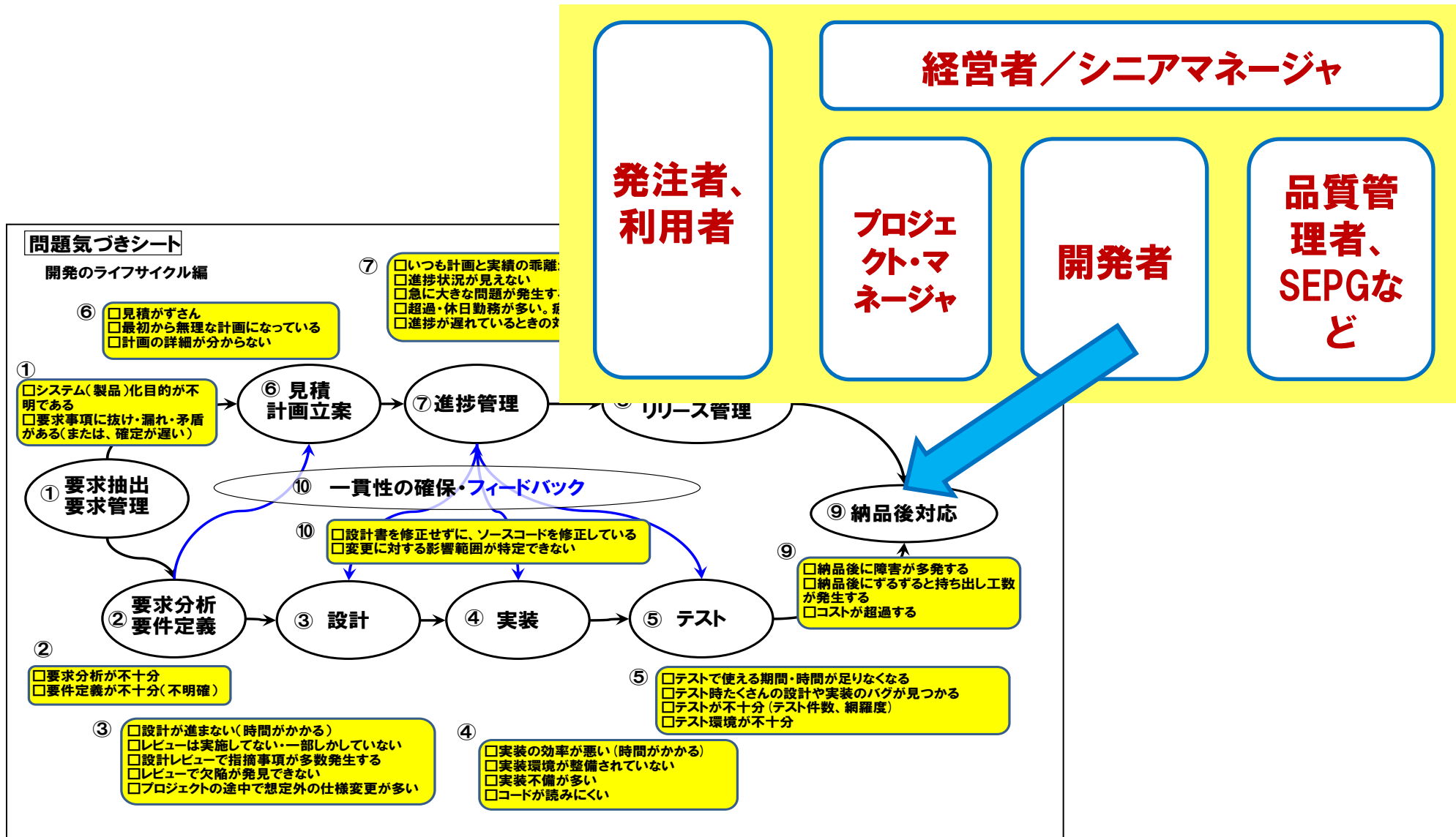
# SPINA<sup>3</sup>CH自律改善メソッドの特色(2/4)

## さまざまな状況・問題・要望(ある組織の事例)



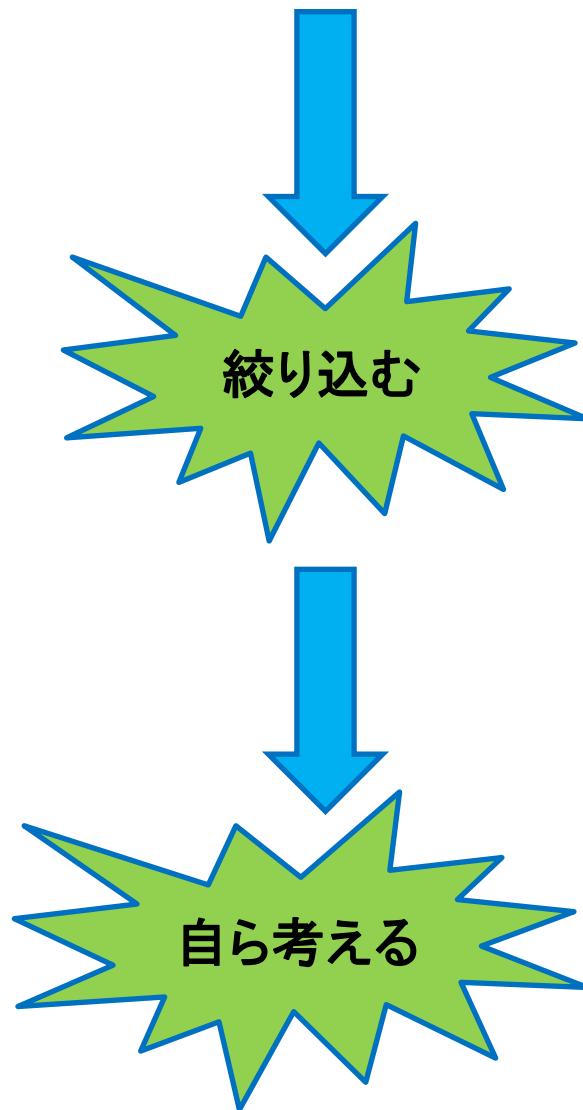
こうした状況を整理して課題を分析していく

# SPINA<sup>3</sup>CH自律改善メソッドの特色(3/4)



# SPINA<sup>3</sup>CH自律改善メソッドの特色(4/4)

## アセスメントモデル: SPEAK-IPA版のプロセス全体像(開発者等向け)



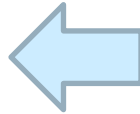
カテゴリ	グループ	プロセス
主ライフサイクルプロセスカテゴリ	顧客-供給者プロセスグループ	<ul style="list-style-type: none"> <li>●P.1.1 取得準備プロセス</li> <li>●P.1.2 供給者選択プロセス</li> <li>●P.1.3 供給者監視プロセス</li> <li>●P.1.4 顧客の受入れプロセス</li> </ul>
	エンジニアリングプロセスグループ	<ul style="list-style-type: none"> <li>●P.2 供給プロセス</li> <li>●P.3.1 要求事項抽出プロセス</li> <li>●P.3.2 システム要求分析プロセス</li> <li>●P.3.3 システムアーキテクチャ設計プロセス</li> <li>●P.3.4 ソフトウェア要求分析プロセス</li> <li>●P.3.5 ソフトウェア設計プロセス</li> <li>●P.3.6 ソフトウェア構築プロセス</li> <li>●P.3.7 ソフトウェア結合プロセス</li> <li>●P.3.8 ソフトウェアテストプロセス</li> <li>●P.3.9 システム結合プロセス</li> <li>●P.3.10 システムテストプロセス</li> <li>●P.5 保守プロセス</li> </ul>
支援ライフサイクルプロセスカテゴリ	支援プロセスグループ	<ul style="list-style-type: none"> <li>●S.1 文書化プロセス</li> <li>●S.2 構成管理プロセス</li> <li>●S.3 品質保証プロセス</li> <li>●S.4 検証プロセス</li> <li>●S.5 妥当性確認プロセス</li> <li>●S.8 問題解決プロセス</li> </ul>
組織ライフサイクルプロセスカテゴリ	管理プロセスグループ	<ul style="list-style-type: none"> <li>●O.1.3 プロジェクト管理プロセス</li> <li>●O.1.4 品質管理プロセス</li> <li>●O.1.5 リスク管理プロセス</li> </ul>
	組織プロセスグループ	<ul style="list-style-type: none"> <li>●O.1.1 組織に関するアライメントプロセス</li> <li>●O.1.2 組織管理プロセス</li> <li>●O.1.6 測定プロセス</li> <li>●O.4.1 人的資源管理プロセス</li> <li>●O.4.2 教育訓練プロセス</li> <li>●O.7 ドメイン技術プロセス</li> </ul>



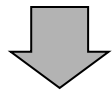
# SPINA<sup>3</sup>CH自律改善メソッドのツール構成

<3つから構成>

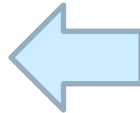
現状問題の気づき



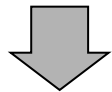
問題気づきシート



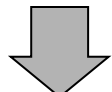
領域の絞り込み



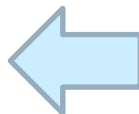
問題分析  
絞り込みシート



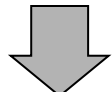
ワークシートの選択



テーマごとの  
ワークシートの記載



改善検討  
ワークシート



改善へ着手

全体として記入シートを使ったプロセス改善のナビゲーション

# どうやって使うか？

STEP1:『問題気づきシート』を使って、開発のライフサイクルで発生している問題を粗くチェック

STEP2:問題の詳細化と因果関係の分析

STEP3:改善の対象を絞る

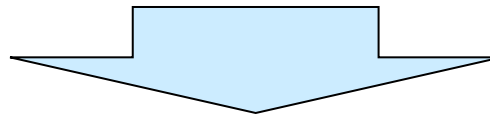
STEP4:該当する改善検討ワークシートを選択する

STEP5:選択された改善検討ワークシートに従い、自らが、やるべき事を考え、記入していく

STEP6:チーム討議や、専門家との討議をして深める

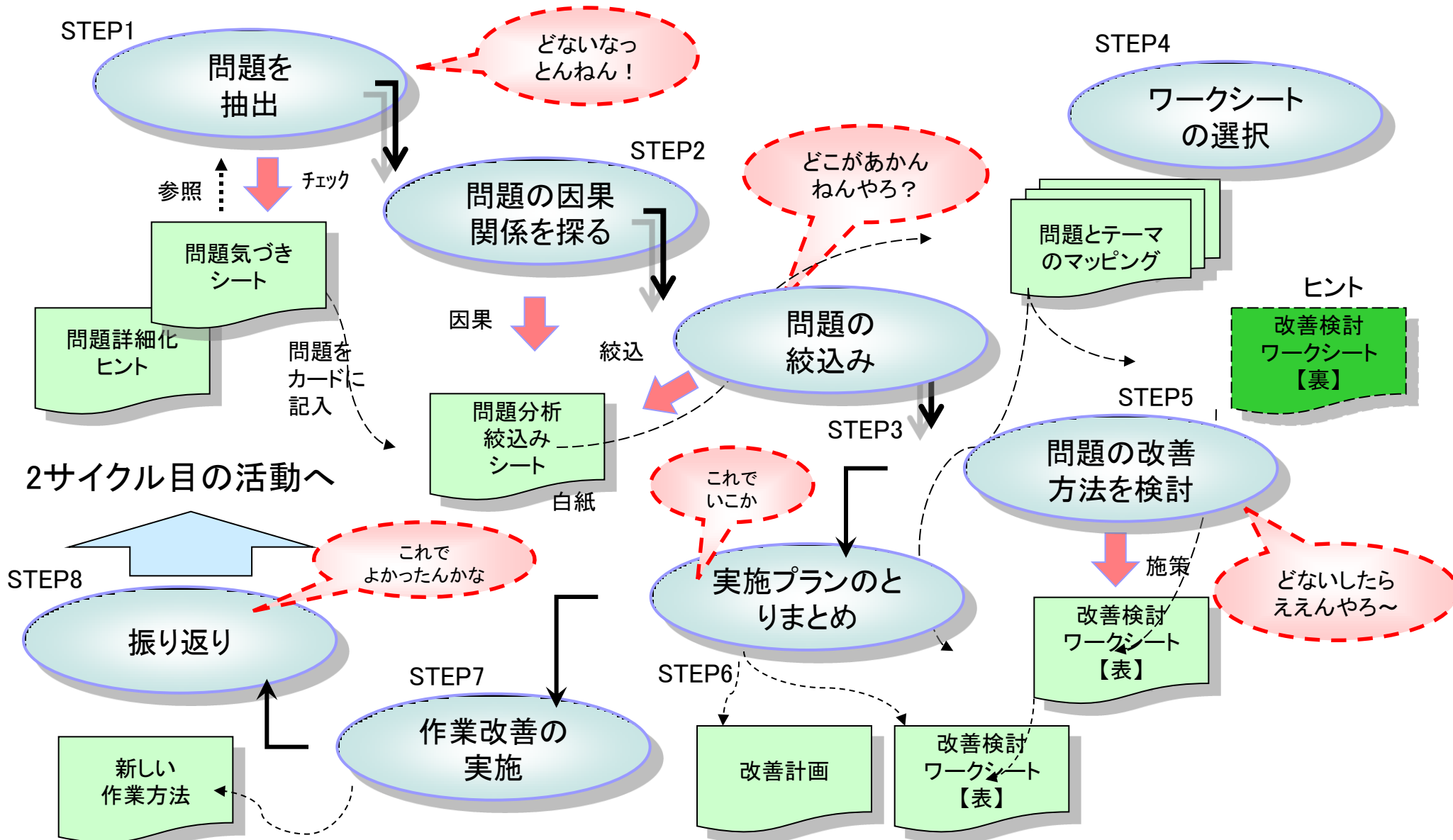
STEP7:ワークシートの検討結果を作業の改善に適用する

STEP8:振り返りを行う



**STEP1～STEP7の結果を、STEP8で再点検し、次のサイクルへつなげる**

# どうやって使うか？



# (参考) SPINA<sup>3</sup>CH実施手順

---

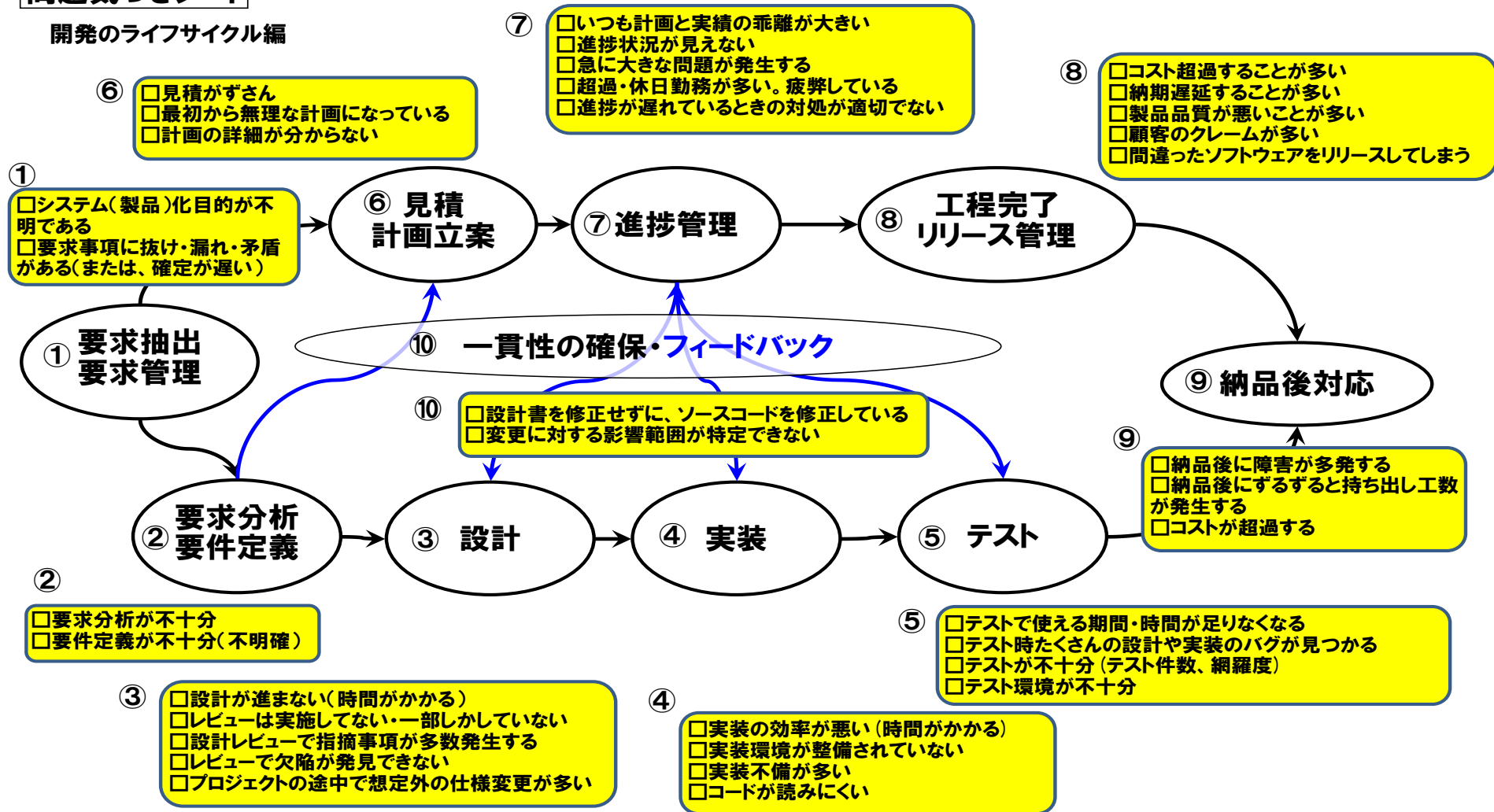
## 準備する主な道具

- 問題気づきシート
- 問題点カード
- 問題詳細化ヒント
- 問題分析絞り込みシート
- 問題とテーマのマッピング
- 改善検討ワークシート

# STEP1 『問題気づきシート』を使って発生している問題を粗くチェック

## 問題気づきシート

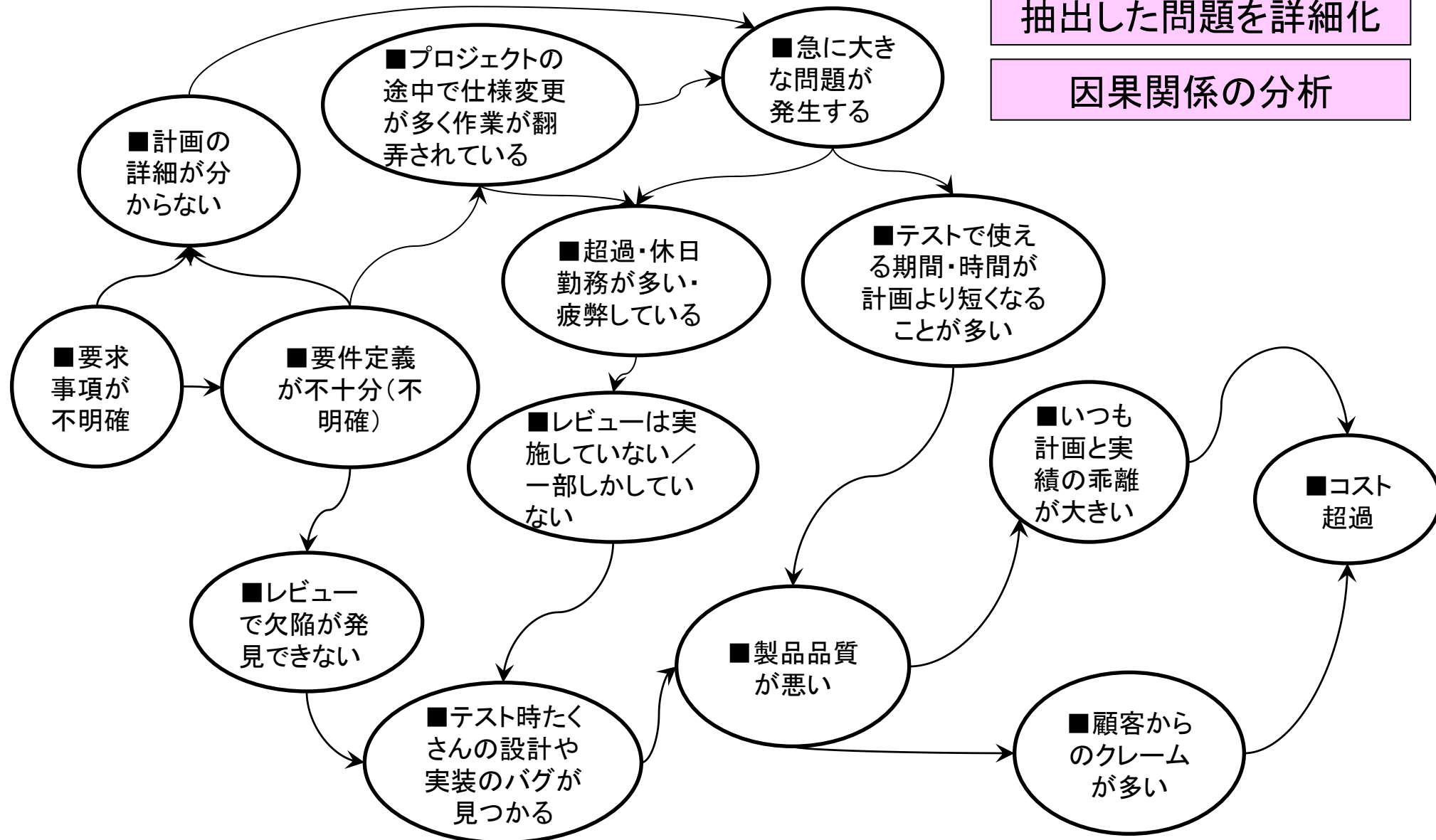
### 開発のライフサイクル編



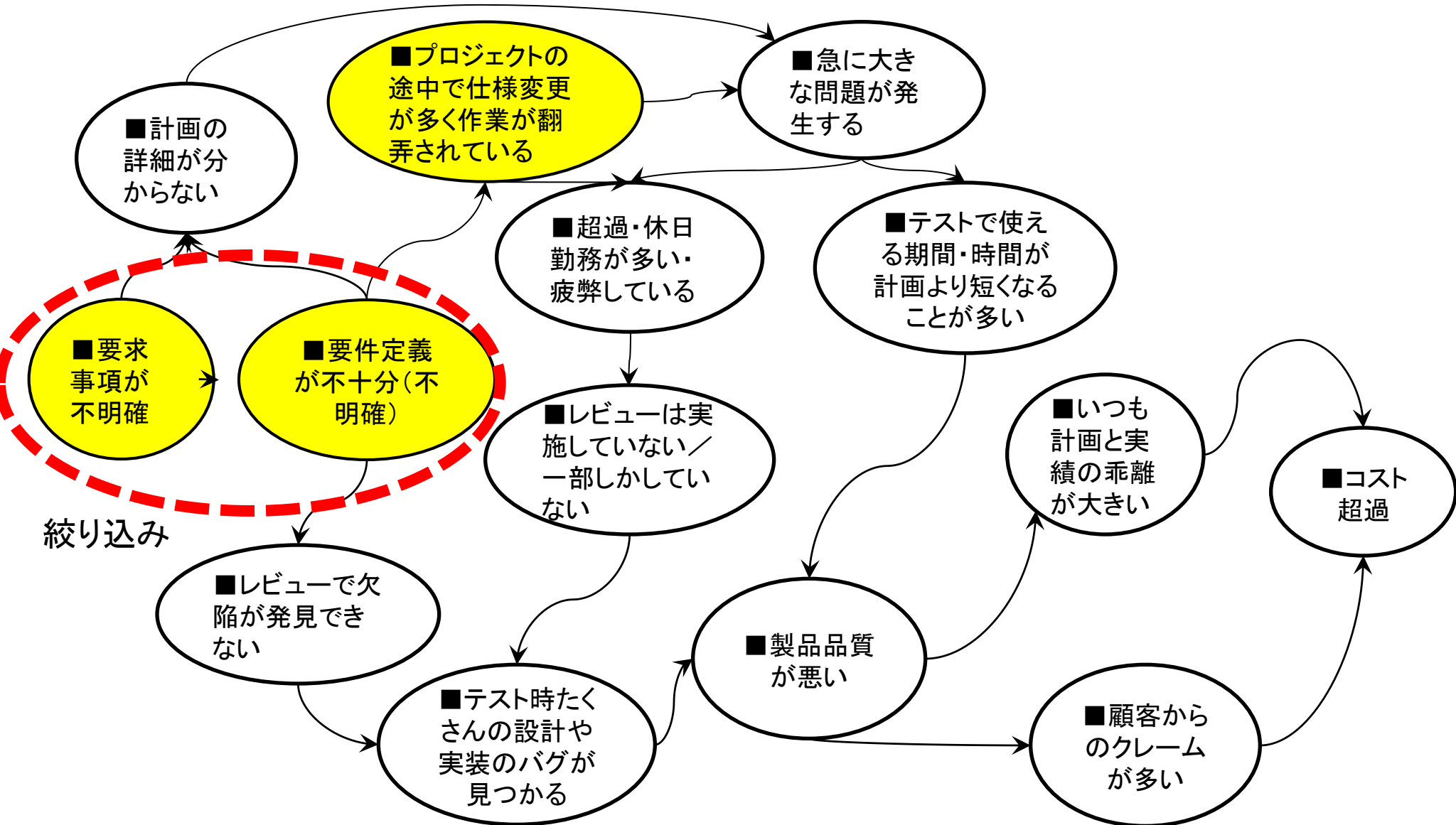
# STEP2 問題の詳細化と因果関係の分析

抽出した問題を詳細化

因果関係の分析



# STEP3 改善対象を絞る



# STEP4 該当する改善検討ワークシートを選択する

## ■開発のライフサイクル編

### 1.顧客要求の明確化

a)システム化目的が不明である	WS-P31-1顧客要求を獲得する仕組みを確立する
	WS-P32-1顧客要求からシステム要求事項にまとめ直す
b)要求事項に抜け・漏れ・矛盾がある (または、確定が遅い)	WS-P31-1顧客要求を獲得する仕組みを確立する
	WS-S1-1作成すべき標準文書と記載すべき内容を明確にする

### 2.要求分析・要件定義

a)要求分析が不十分	WS-JRev-1プロジェクトの作業成果物について利害関係者とレビューを実施する。
	WS-S5-2妥当性確認(顧客が本来やりたいことが実装できているかを確認すること)実施する
b)要件定義が不十分(不明確)	WS-P32-1顧客要求からシステム要求事項にまとめ直す
	WS-P31-1顧客要求を獲得する仕組みを確立する
	WS-S1-1作成すべき標準文書と記載すべき内容を明確にする
	WS-S5-2妥当性確認(顧客が本来やりたいことが実装できているかを確認すること)実施する

### 3.設計

a)設計が進まない(時間がかかる)	WS-P33-2システムアーキテクチャ設計の適切性を確認する
	WS-P34-2ソフトウェア要求事項の適切性を確認する
b)レビューは実施していない・一部しかしていない	WS-JRev-1プロジェクトの作業成果物について利害関係者とレビューを実施する。
c)設計レビューで指摘事項が多数発生する	WS-JRev-1プロジェクトの作業成果物について利害関係者とレビューを実施する。
	WS-P33-1システム要求を満たすシステムアーキテクチャ設計を実施する
	WS-P35-1ソフトウェア要求を満たすソフトウェア設計を実施する
	WS-P35-3ソフトウェア設計を元にソフトウェア詳細設計を実施する
d)レビューで欠陥が発見できない	WS-JRev-1プロジェクトの作業成果物について利害関係者とレビューを実施する。



# STEP5 改善検討ワークシートを作成する

## 選択した領域に該当するワークシートを作成する過程を通じて、自ら改善手段を導き出す。

改善検討ワークシート	
1. テーマと課題 ①取り上げたテーマ	5. この問題にどのように対処したいか ※例えば裏のソリューション例を参照して、考えてみてください
②このテーマが解決すると何が良くなるか	<b>対応方法</b>
2. この課題について、現在どのように行っているのか	
<b>現状のやり方</b>	
3. 現状のやり方でうまくいっているところと悪いところ ※現在のやり方に対して、どこがうまくいっていて、どこが悪いが記載する	6. 上記対処をするための懸念事項 ※例えば、対処実施の際の、制約事項、コスト、他への影響など
<b>現状のやり方でうまくいっているところと悪いところ</b>	<b>懸念事項</b>
4. 現在のやり方の不十分点・改善すべき点はどこか ※問題の分析・絞り込み時に考慮した改善箇所と関連づけると良い	7. 上記対処をするための留意事項 ※例えば、必要な協力者とか関係者への支援の要望など
<b>改善点</b>	<b>留意事項</b>

改善検討ワークシート (WS-S2-1)	
1. テーマと課題 ①取り上げたテーマ	6. ソリューション例1 ・構成管理を実施するためには、例えば、ベースライン管理の作成計画、特定のベースラインに含まれる作業成果物のバージョン、一貫性を保つためのリリース管理の作成、ベースラインを特定するための命名規約作成などの活動がある ・構成管理は、組織やプロジェクトによって定義が異なる。そのため、組織やプロジェクトでの構成管理の手順を、研修・OJT・オンラインセッションで周知したほうがよい ・バックアップテープの保管やプログラムソースの目付、バージョン、各版へのコメントの記録、ファイルの登録(チェックイン)、取出し(チェックアウト)、差分情報の表示、ソフトウェア構成変更の制御などを構成管理と定義している例もある
②このテーマが解決すると何が良くなるか	<b>テーマに沿った解決事例</b>
2. テーマを要請しなかった場合、起るであろう問題	
3. テーマを要請するために必要な実施事項(典型的な活動)	7. 参照するとよい、他の改善検討ワークシート
4. 実現には要請することが難しい点	8. 参考文献(SPEAK-IPA版のプロセス、その他の資料文献)

改善検討ワークシート

<p>1. テーマと課題</p>	<p>5. この問題にどのように対処したいか</p>
<p>①取り上げたテーマ</p>	<p>※例えば裏のソリューション例を参照して、考えてみてください</p>
<p>②このテーマが解決すると何が良くなるか</p>	
<p>2. この課題について、現在どのように行っているのか</p>	
<p style="text-align: center;"><b>現状のやり方</b></p>	<p style="text-align: center;"><b>対応方法</b></p>
<p>3. 現状のやり方でうまくいっているところと悪いところ</p>	<p>6. 上記対処をするための懸念事項</p>
<p>※現在のやり方に対して、どこがうまくいっていて、どこが悪いか記載する</p>	<p>※例えば、対処案実施の際の、制約事項、コスト、他への影響など</p>
<p style="text-align: center;"><b>現状のやり方でうまくいっているところと悪いところ</b></p>	<p style="text-align: center;"><b>懸念事項</b></p>
<p>4. 現在のやり方の不十分点・改善すべき点はどこか</p>	<p>7. 上記対処をするための留意事項</p>
<p>※問題の分析・絞り込み時に考慮した改善箇所と関連づけると良い</p>	<p>※例えば、必要な協力者とか関係者への支援の要望など</p>
<p style="text-align: center;"><b>改善点</b></p>	<p style="text-align: center;"><b>留意事項</b></p>

## 改善検討ワークシート

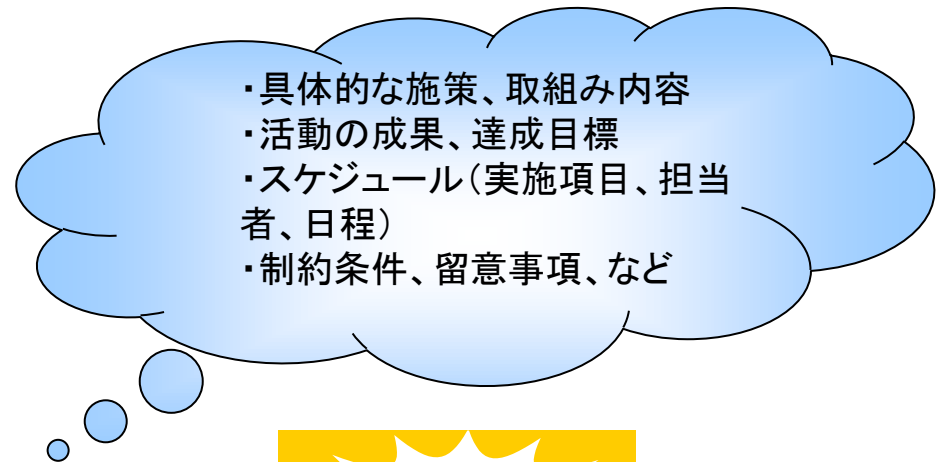
(WS-S2-1)

<b>1. テーマと課題</b>	<b>5. ソリューション例1</b>
<p>①取り上げたテーマ</p> <p>構成管理の戦略やしきみがある</p> <p>②このテーマが解決すると何が良くなるか</p> <ul style="list-style-type: none"> <li>・構成管理の戦略やしきみを確立することで、構成管理の3要素である「バージョン管理」「ベースライン管理」「変更管理」についてのルール作りと運用を徹底できる</li> <li>・プロジェクトのライフサイクルで生成される構成要素をベースライン管理することでデグレードやファイルの取り違えを起こすことなくリリースにつなげることができる</li> <li>・リリースバージョンにどのような機能が盛り込まれているかが適切に管理できる</li> </ul>	<ul style="list-style-type: none"> <li>・構成管理を実施するためには、例えば、ベースライン管理の作成計画、特定のベースラインに含まれる作業成果物のバージョン、一貫性を保つトレーサビリティマトリクス作成、ベースラインを特定するための命名規約作成などの活動がある</li> <li>・構成管理は、組織やプロジェクトによって定義が異なる。そのため、組織やプロジェクトでの構成管理の手順を、研修・OJT・オリエンテーションで周知したほうがよい</li> <li>・バックアップテープの保管やプログラムソースの日付、バージョン、各版へのコメントの記録、ファイルの登録(チェックイン)、取出し(チェックアウト)、差分情報の表示、ソフトウェア構成変更の制御などを構成管理と定義している例もある</li> </ul>
<p><b>2. テーマを実施しなかった場合、起こるであろう問題</b></p> <ul style="list-style-type: none"> <li>・成果物の構成要素の一貫性と追跡可能性を維持できなくなる</li> <li>・仕様変更や保守を管理する際に、ドキュメントの内容を分析し、必要な変更事項を決定し、該当するプログラムを変更する等の手順が踏めなくなる</li> <li>・ソフトウェア開発の作業にデグレード、ファイルの取り違えなどの事故や不具合が発生し、非効率な作業、不正確な情報、不</li> </ul>	<p><b>6. ソリューション例2</b></p> <p>ツールを導入しない場合、チェックイン・チェックアウトなどの機能をフォルダー管理により代替する方法もある。</p> <p>例えば、プロジェクトの共用フォルダーにパスワードを設け、構成管理担当者とプロジェクトメンバー以外は「書き込み禁止」することで、プログラム担当者が、作成して組み込むことはなくなり、デグレードや</p>
<h1 style="background-color: cyan; color: black; padding: 10px;">テーマに沿った解決事例</h1>	
<p><b>3. テーマを実現するために必要な実施事項(典型的な活動)</b></p> <ul style="list-style-type: none"> <li>・構成管理の方針、作業一覧、作業生産物、構成管理の責任者、定期的な監査のタイミング、を含む構成管理活動の戦略を立てる</li> <li>・仕様書やソースコードなど、構成管理の対象となる作業成果物を明確にする</li> <li>・プロジェクトの目標、リスクなど複数の制御レベルを管理する仕組みを確立する</li> <li>・構成ベースラインの一貫性を維持するため、構成管理の監査を定期的に行う</li> </ul>	<p><b>7. 参照するとよい、他の改善検討ワークシート</b></p> <ul style="list-style-type: none"> <li>・WS-S2-2プロジェクトで作成すべき成果物(中間成果物と納入成果物)と作成時期が決まっている</li> <li>・WS-S2-3プロジェクトで作成すべき成果物(中間成果物と納入成果物)の変更管理がされている</li> <li>・WS-S2-4プロジェクトで作成すべき成果物(中間成果物と納入成果物)の間の一貫性を保つ仕組みがある</li> </ul>
<p><b>4. 現実には実現することが難しい点</b></p> <ul style="list-style-type: none"> <li>・仕様変更などで、ソースコードのみを修正してしまい、設計書やテスト仕様などの一貫性を保つ必要がある関連の成果物を修正しない場合がある</li> <li>・大規模プロジェクトでは機能追加などで仕様変更が多くなり、成果物の構成要素も増加する。そのため、多くのモジュールの世代を管理したり、一世代前に戻してリンクし直すなど、変更が管理されておらず、成果物のバージョンや変更の履歴がとれない</li> <li>・構成管理は、要件定義書、設計書、プログラム、テスト仕様などを横断的に管理するべきである。しかし、実際には関連するドキュメントを最新化して同時に構成要素をベースライン管理する煩雑な作業の手間を惜み、プログラムモジュールのバージョンを管理することに精一杯である</li> </ul>	<p><b>8. 参考文献(SPEAK-IPA版のプロセス、その他の資料文献)</b></p> <ul style="list-style-type: none"> <li>・エドワード・ヨードン著、松原 友夫訳、ソフトウェア管理の落とし穴 —アメリカの事例に学ぶ—、トッパン、1993年</li> <li>・F.P.ブルックス著、山内正彌訳、ソフトウェア開発の神話、企画センター、1977年</li> <li>・東秀和(データプロセス)、@It自分戦略研究所、第4回 Subversionで簡単・確実にファイルを構成管理、 <a href="http://jibun.atmarkit.co.jp/lskill01/rensai/tool10/04/01.html">http://jibun.atmarkit.co.jp/lskill01/rensai/tool10/04/01.html</a></li> </ul>

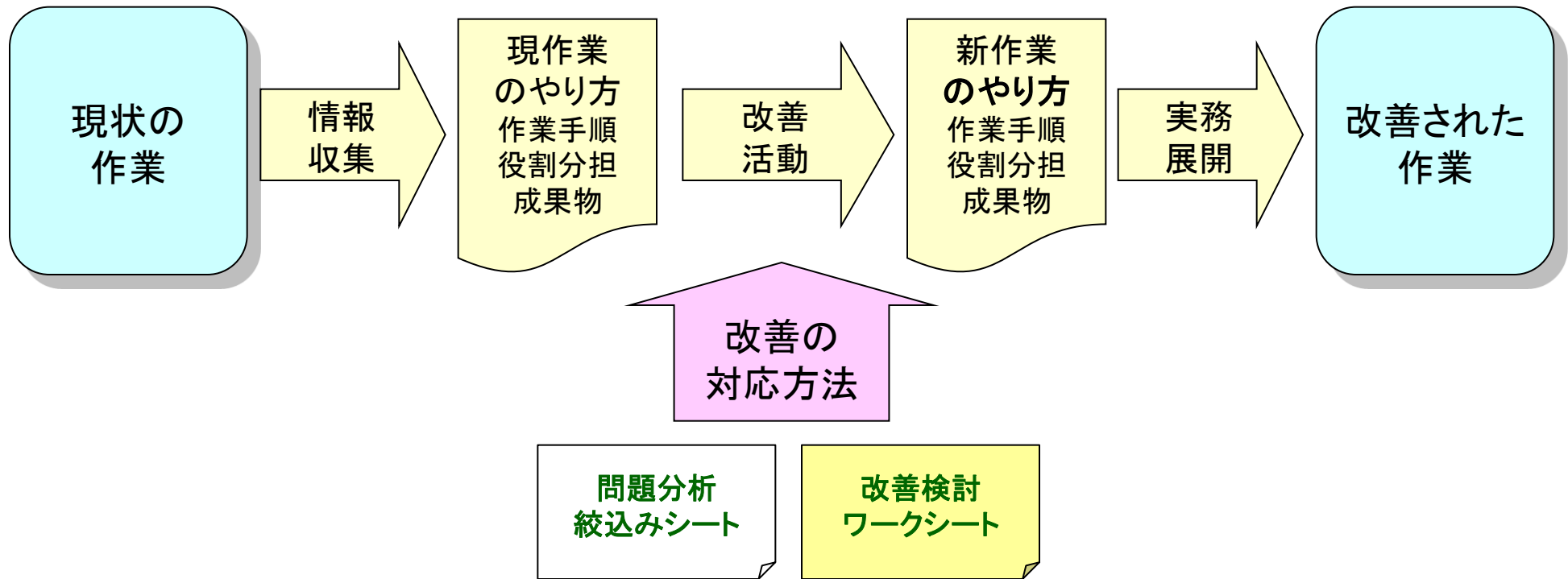
## STEP6 チーム討議や、専門家との討議をして深める

# 改善検討ワークシートの内容をもとに**具体的な実施プラン**を取りまとめる。

改善検討ワークシート	
1. テーマと課題 ①取り上げたテーマ	5. この問題にどのように対処したいか ※例えば、要のソリューション例を参照して、考えてみてください
②このテーマが解決すると何が良くなるか	<b>対応方法</b>
2. この課題について、現在どのように行っているのか	
<b>現状のやり方</b>	
3. 現状のやり方でうまくいっているところと悪いところ ※現在のやり方に対して、どこがうまくいっていて、どこが悪いが記載する	6. 上記対処をするための懸念事項 ※例えば、対処案実施の際の、制約事項、コスト、他への影響など
<b>現状のやり方でうまくいっているところと悪いところ</b>	<b>懸念事項</b>
4. 現在のやり方の不十分点・改善すべき点はどこか ※問題の分析・絞り込み時に考慮した改善箇所と関連づけると良い	7. 上記対処をするための留意事項 ※例えば、必要な協力者とか関係者への支援の要望など
<b>改善点</b>	<b>留意事項</b>

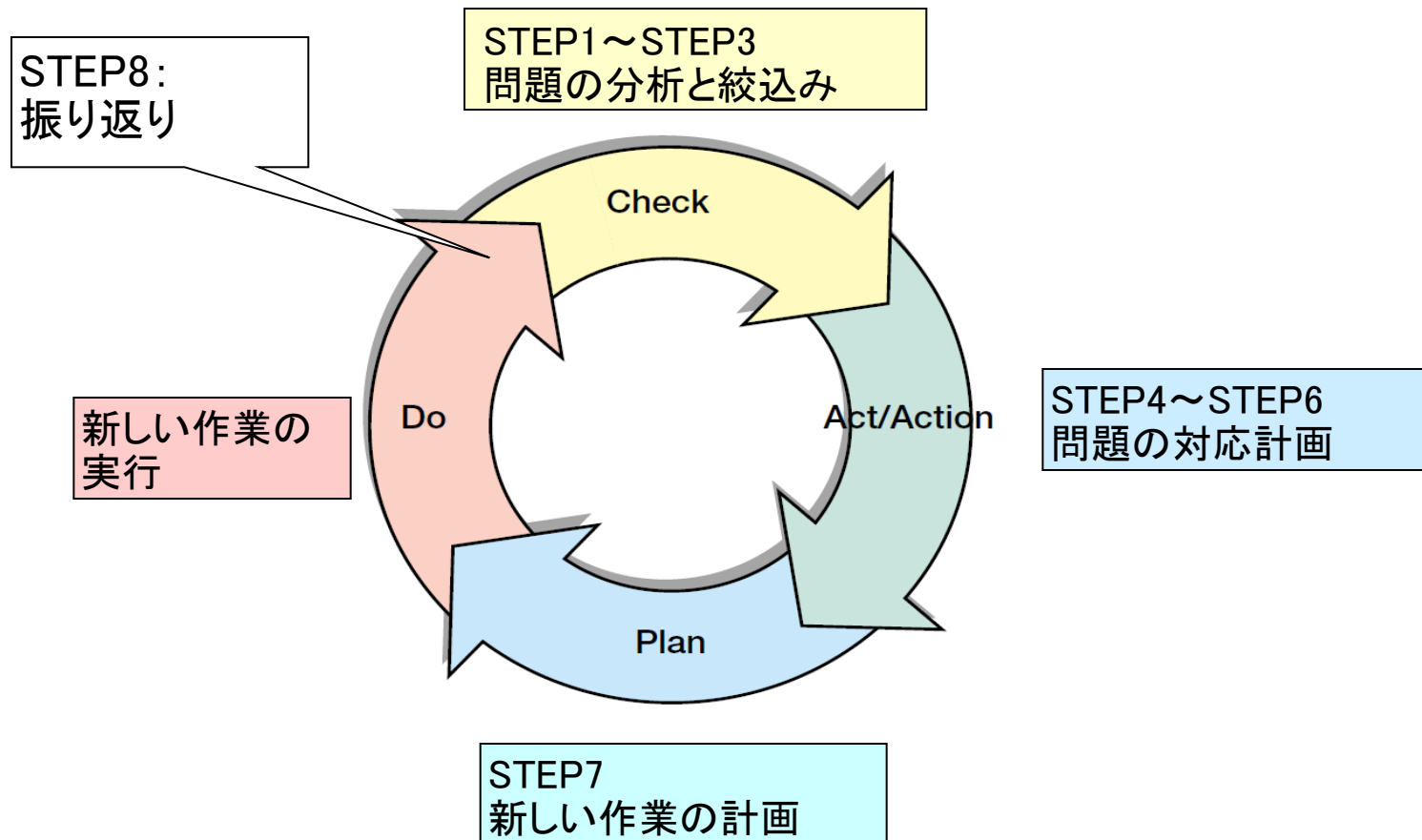


現作業のやり方を明確化し、これに対し改善の「対応方法」を適用し、新作業のやり方を創出する。

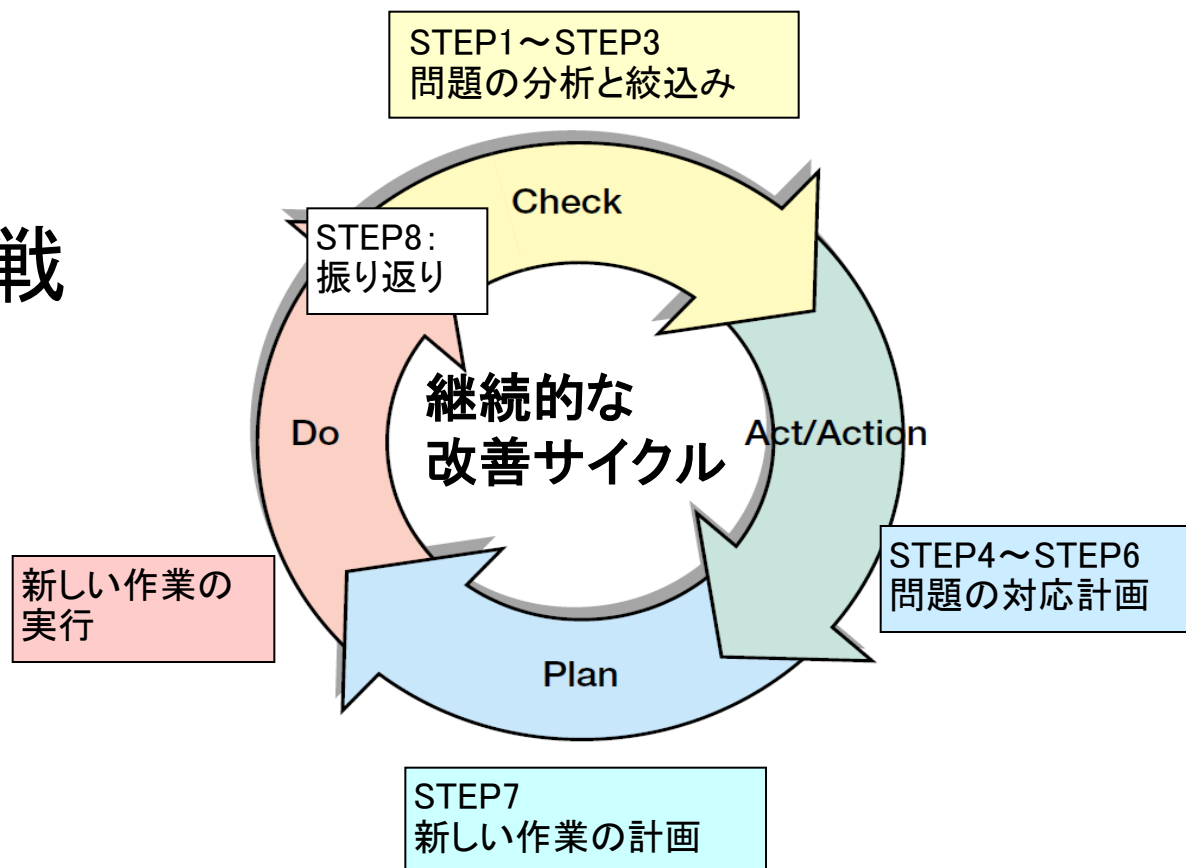


# STEP8:振り返り

改善活動を振り返り、実施してきた内容の良かったところ悪かったところをまとめる。この内容をもとに、(STEP1～STEP7)で実施した作業のやり方を再点検し、作業の進化を確認する



- さらなる改善へ
- 組織展開
- 新たな課題に挑戦



ご清聴ありがとうございました