

TPI Nextを活用した
チームメンバーの問題意識から
始めるテストプロセス改善
【導入時：改善計画立案編】

Test Engineer's Forum 北海道(TEF道)
安達賢二(HBA)

今回の事例

- JaSST'17北海道で発表～テストチームのプロセス改善をTPI Nextを使ってやってみよう！と思ったリーダ中山さんの事例の詳細。
- TPI Next日本語書籍を一読した中山さん。セルフアセスメントをやってみた。

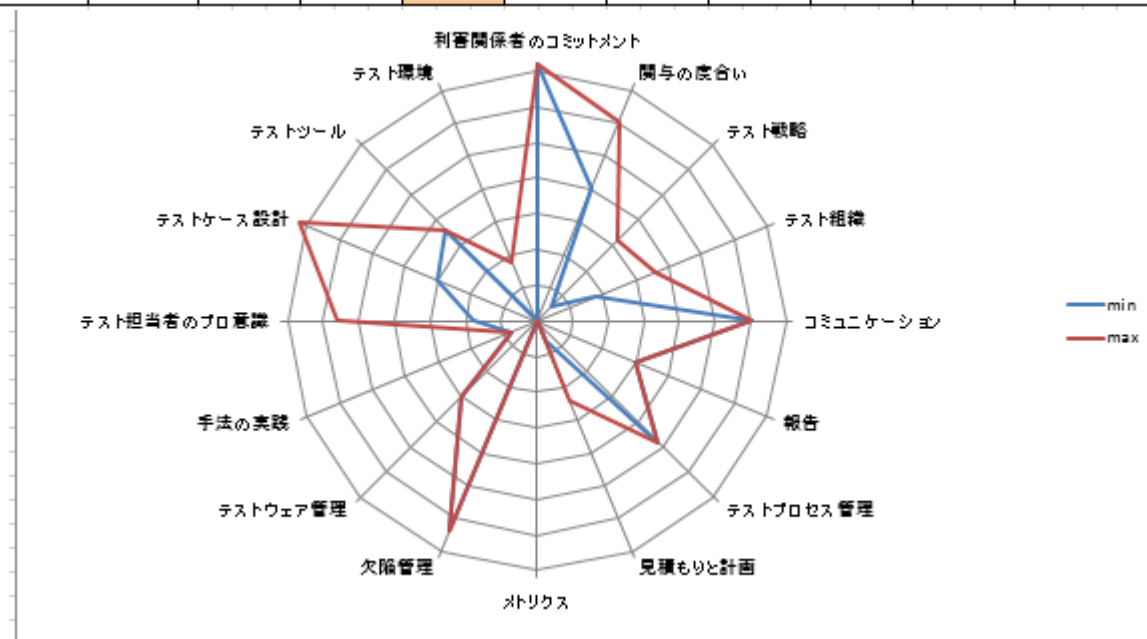
プロセスモデルに精通する有識者は多くはない
組織やチームの改善活動も積極的ではない
そのような背景でも「何とかしたい！」と思っているリー
ダやエンジニアが多いのではないかな？

TPI Nextチェック項目の一部

SR	1.利害関係者のコミットメント	
	初期レベル	
	コントロールレベル	
	1	利害関係責任者を決定し（必ずしも文書化の必要はない）
	2	テストリソースに対する予算は、利害関係責任者が認める
	3	利害関係者はコミットしたリソースを実際に手配している。
	4	利害関係責任者は、文書化されたプロダクトリスク分析
	効率化レベル	
	1	関連するすべての利害関係者を定義して（必ずしも文書化する）。
	2	利害関係者は、積極的にテストプロセスやテスト対象の品
	3	利害関係者は、テストプロセスに影響を与える側面について、対象のテスト作業へのリリース順序やプロジェクトスコープが
	最適化レベル	
	1	テストプロセスを改善するときは、提供するリソースへの学習
	2	利害関係者は、ソフトウェア開発や要件管理といった仕事
	3	利害関係者による仕事の進め方をテストプロセスの要求に
	双方によって評価されている。	
SR	2.関与の度合い	
	初期レベル	
	コントロールレベル	
	1	最初のテスト活動として、テストの任務、スコープ、取り組み
	2	テスト活動をプロジェクトのクリティカルパスにしないよう、テ
	3	プロジェクト計画において、テストプロセスとその他のプロセス
	4	計画に関与している。
	最適化レベル	
	1	テスト担当者、プロジェクト全体のテストリソースの
	2	テスト活動の進捗状況を定期的に確認し、必要に応じて
TM	11.テストウェア管理	
	初期レベル	
	コントロールレベル	
	1	テストベースとテスト対象およびすべてのテストウェアを、名前とバージョンで特定している。
	2	各テストケースを、テストベース文書に明白な方法で関連付けている。
	3	テストチームは、テストウェアの管理下のすべてのアイテムにアクセスできる。
	4	テストウェア、テストベース、テスト対象の扱い方が明確に規定され、テストチームに伝えられている。
	効率化レベル	
	1	テストベースとテスト対象およびすべてのテストウェアを、名前とバージョンで参照できる。
	2	テストケースと要件のトレーサビリティが確保されている。
	3	テストウェア管理は、論理的な補充構造と、役割および権限の構造によって支えられている。
	最適化レベル	
	1	テストプロジェクト終了後にどのテストウェアを保持するかをテストプロジェクト開始時に合意し、テストプロジェクト実施中に見直している。
	2	再利用に備えたテストウェア保持に関するガイドラインが入手可能な状態にあり、テストウェアの再利用を測定している。
	3	プロジェクト終了時に保守に引き渡すテストウェアが、保守されないテストウェアと容易に分離できる。
TP	12.手法の実践	
	初期レベル	
	コントロールレベル	
	1	テストプロセスは、明文化されたテスト手法に従っている。テスト手法には、一連のテスト活動、テストプロジェクトの成果となるテストプロダクト、作業の途中で発生する追加要求について記述している。
	2	テスト手法は、プロジェクトが用いている開発手法に適合している。
	3	テストプロジェクトにとって、実装したテスト手法が実用的なものであると認識されている。
	効率化レベル	
	1	テスト手法には、すべてのテスト活動に関するゴールと役割、および用いるべき技法と事前条件が明文化されている。
	2	完全かつ包括的なテスト計画のフォーマットが、テスト手法の二重チェックを提供されている。
	3	テスト手法は、プロジェクトの進捗状況を定期的に確認し、必要に応じて

当初のセルフアセスメント結果

		初期レベル	コントロール レベル				効率化 レベル				最適化 レベル		
1	利害関係者のコミットメント		1	2	3	4	1	2	3		1	2	3
2	関与の度合い		1	2	3	4	1	2	3		1	2	
3	テスト戦略		1	2	3	4	1	2	3		1	2	
4	テスト組織		1	2	3	4	1	2	3	4	1	2	3
5	コミュニケーション		1	2	3	4	1	2	3		1	2	
6	報告		1	2	3	4	1	2	3		1	2	
7	テストプロセス管理		1	2	3	4	1	2	3		1	2	
8	見積もりと計画		1	2	3	4	1	2	3	4	1	2	3
9	メトリクス		1	2	3	4	1	2	3	4	1	2	
10	欠陥管理		1	2	3	4	1	2	3	4	1	2	3
11	テストウェア管理		1	2	3	4	1	2	3		1	2	3
12	手法の実践		1	2	3	4	1	2	3	4	1	2	
13	テスト担当者のプロ意識		1	2	3	4	1	2	3	4	1	2	3
14	テストケース設計		1	2	3	4	1	2	3	4	1	2	3
15	テストツール		1	2	3	4	1	2	3	4	1	2	3
16	テスト環境		1	2	3	4	1	2	3	4	1	2	3



アセスメント結果は出たものの・・・

- 評価結果は正しいのか？実態に合っているのか？不安が・・・。
- 多くのNGが・・・どこから手をつけるべきか？？
 - モデルが推奨するクラスタAから順に・・・
 - ✓それだと改善が実感できず、長引いて続かなくなるかもよ
 - 中山さんが普段から問題ではないかと気になっていた「テスト環境」に着目して・・・
 - ✓それがメンバーの人たちが解消したい1番の困り事なのかな？
- **じゃあどうすればいいの？？？**

プロセスモデルベース改善の問題点

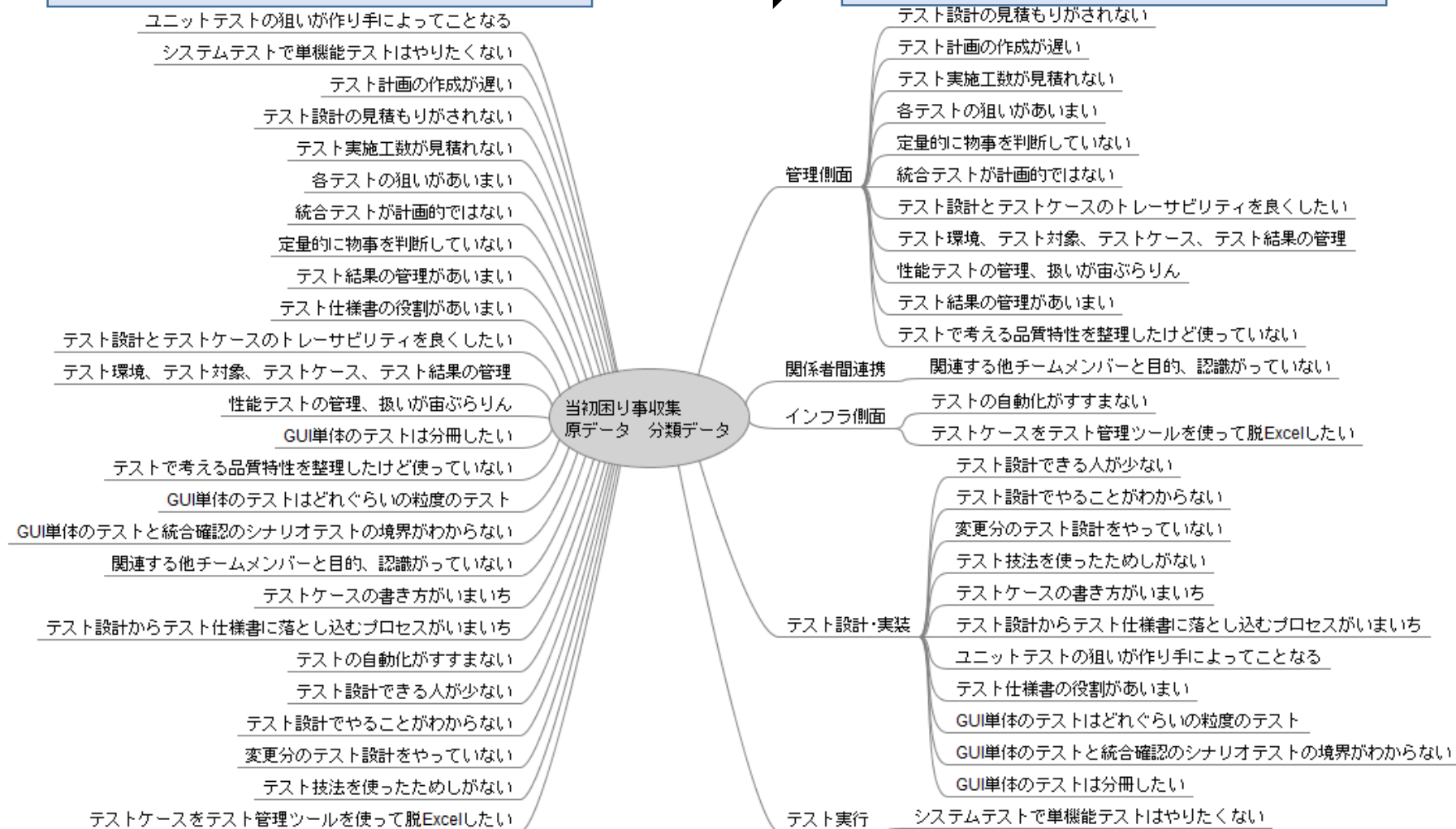
- 重たい
 - 例えばTPI Next: 16プロセスエリア143プラクティス
 - みんなで評価するのは理想だが現実的に無理な場合が多い
 - 第三者や代表者が評価する……他人事化しやすい
- モデルが難解
 - 個別のプラクティスがなかなか読み解けない
 - 対象は何？どこまでやれるとOK？わかるようでわかりにくい
- 評価結果の意味がわからない
 - 大抵は少しのOKとたくさんのNG・・・モチベーションダウン
 - その意味は？？何から手をつけるべき？判断の拠り所は？
- 取り組んだ結果の効果が実感しにくい
 - 何を目指して取り組めばいいのか？
 - 考えるのが面倒になるとモデル適合を目標にしがち・・・それは誰が、何がうれしいの？

**プロセスモデルをしっかりと理解してから活用しないと
思わぬ怪我(迷走・頓挫・自然消滅等)をしてしまう**

メンバーからふりかえりコメントを収集・分類

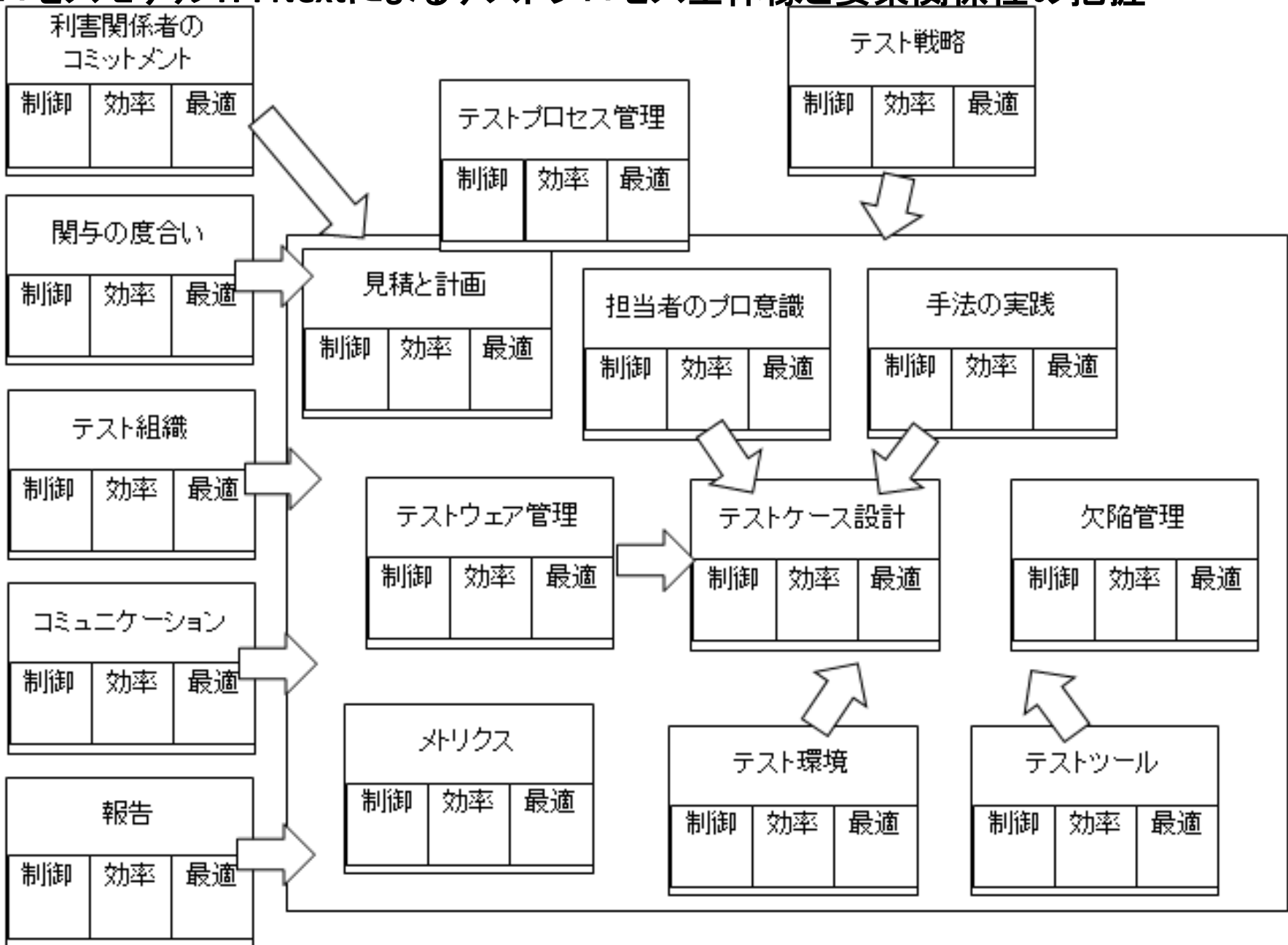
実際のふりかえり結果

整理・分類結果

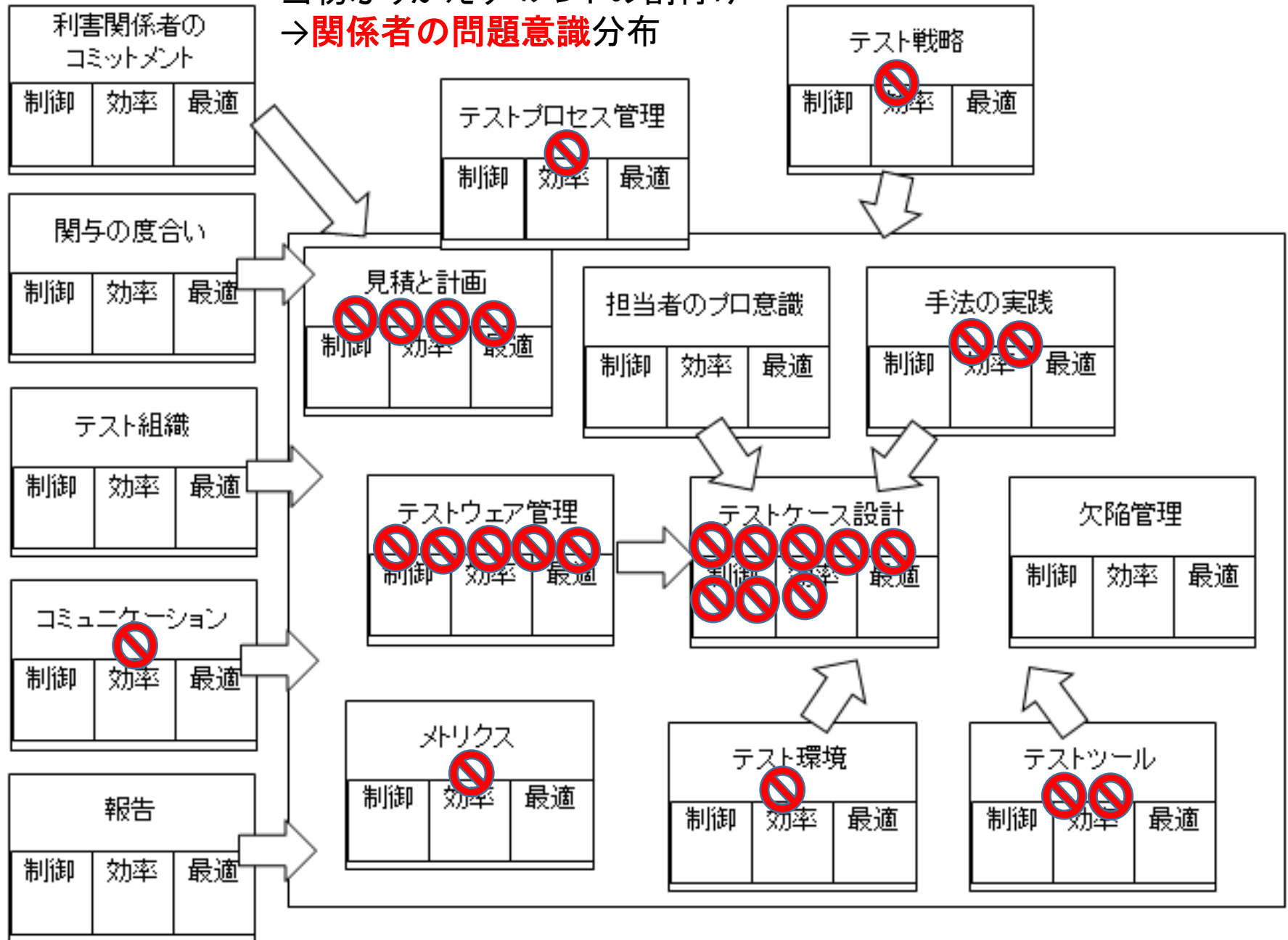


地図に疑問を持ったら現地を直接調べよう

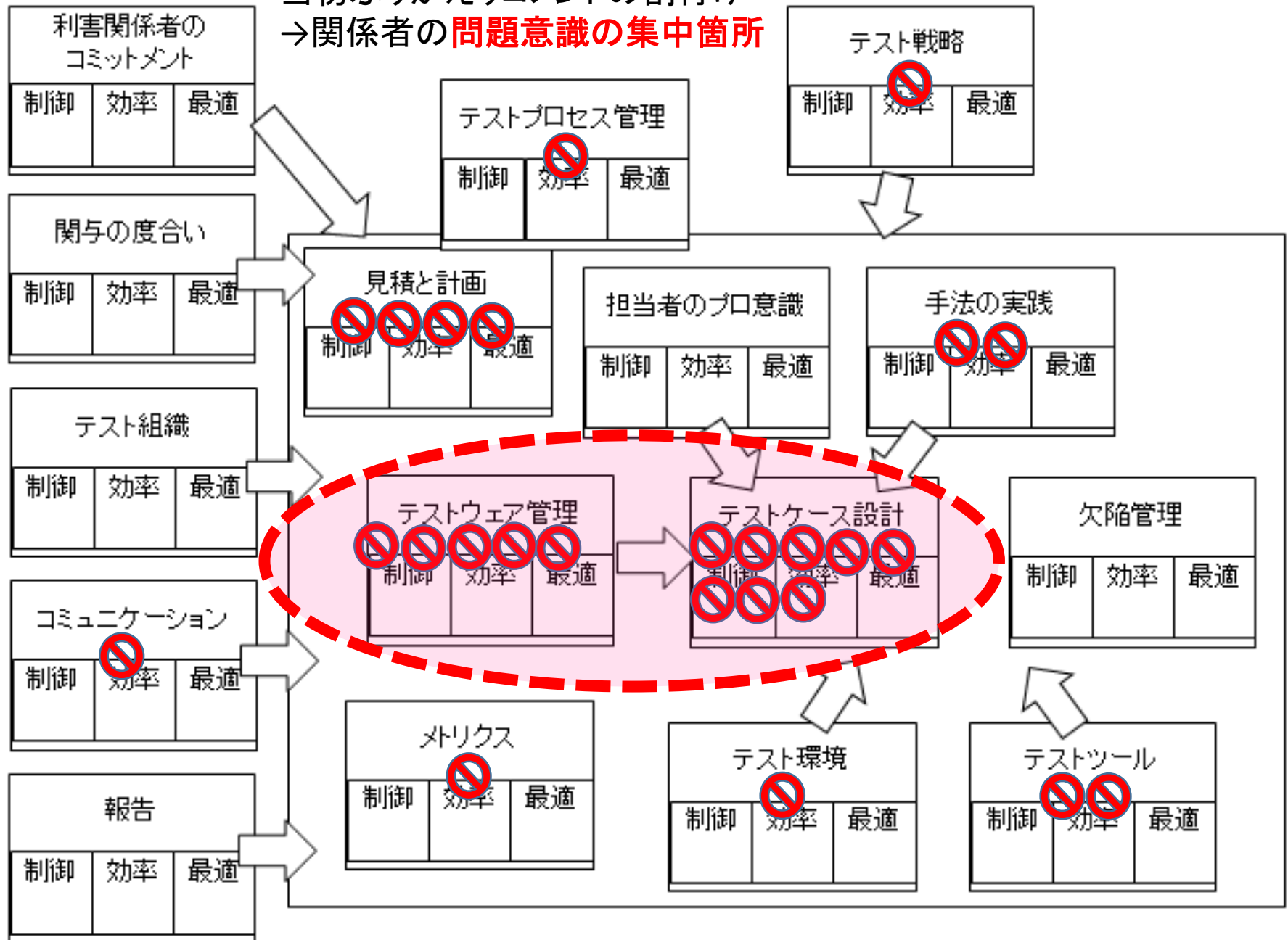
プロセスモデルTPI Nextによるテストプロセス全体像と要素関係性の把握



当初ふりかえりコメントの割付け
→関係者の問題意識分布



当初ふりかえりコメントの割付け
→関係者の**問題意識の集中箇所**



実際に存在する品質・コスト・期間(QCD)問題関連事象・困り事の把握

管理側面

- テスト設計の見積もりがされない
 - 見積りされないことでどのような困り事が発生している？
- テスト計画の作成が遅い
 - 計画立案スピードのこと？計画が出来上がるタイミングのこと？
- テスト実施工数が見積れない
 - その結果どのような困り事が発生している？
- 定量的に物事を判断していない
 - その結果どのような困り事が発生している？
- 統合テストが計画的ではない
 - 具体的にどのような状態？
 - 状況に適切なテストの組み立てになっていない（戦略的ではない）
 - その結果どのような困り事が発生している？
- 性能テストの管理、扱いが宙ぶらりん
 - 管理とは具体的に何？宙ぶらりんとは具体的にどのようなこと？
 - その結果どのような困り事が発生している？
- テスト結果の管理があいまい
 - 管理とは具体的に何？あいまいとは具体的に何？
 - その結果どのような困り事が発生している？
- 各テストの狙いがあいまい
 - 狙いが曖昧とは具体的にどのような状態？
 - システムテストで何を確認したいのか、GUIテストで何を確認したいのか
 - その結果どのような困り事が発生している？
- テストの境界に対して人によって判断が異なるため、重複テストやテスト漏れだけは防ぐようにしている（手間がかかっている）
- テスト設計とテストケースのトレーサビリティを良くしたい
 - どのトレーサビリティがどのようによくない？
 - “テスト仕様書の役割があいまい”に関連する
 - 元から存在する機能に対するテストケースの粒度がバラバラ
 - スパゲッティテスト仕様状態
 - その結果どのような困り事が発生している？
- テスト環境、テスト対象、テストケース、テスト結果の管理
 - 具体的にどのような状態？
 - “テスト仕様書の役割があいまい”に関連する
 - 過去のテスト結果が残っているがテスト環境情報が残っていないもの
 - 定期的に回帰テスト実施前にNG項目を把握して作戦を立てる際に役立つ
- 結果的にすべてをテスト実施することになる
 - テストで考える品質特性を整理したけど使っていない
 - 使わないことでどのような困り事が発生している？
 - 狙いがあいまいの背景のために作成したが終了

関係者間連携

- 制御メンバーと目的、認識がっていない
 - その結果どのような困り事が発生している？

インフラ側面

- テストの
- Exce

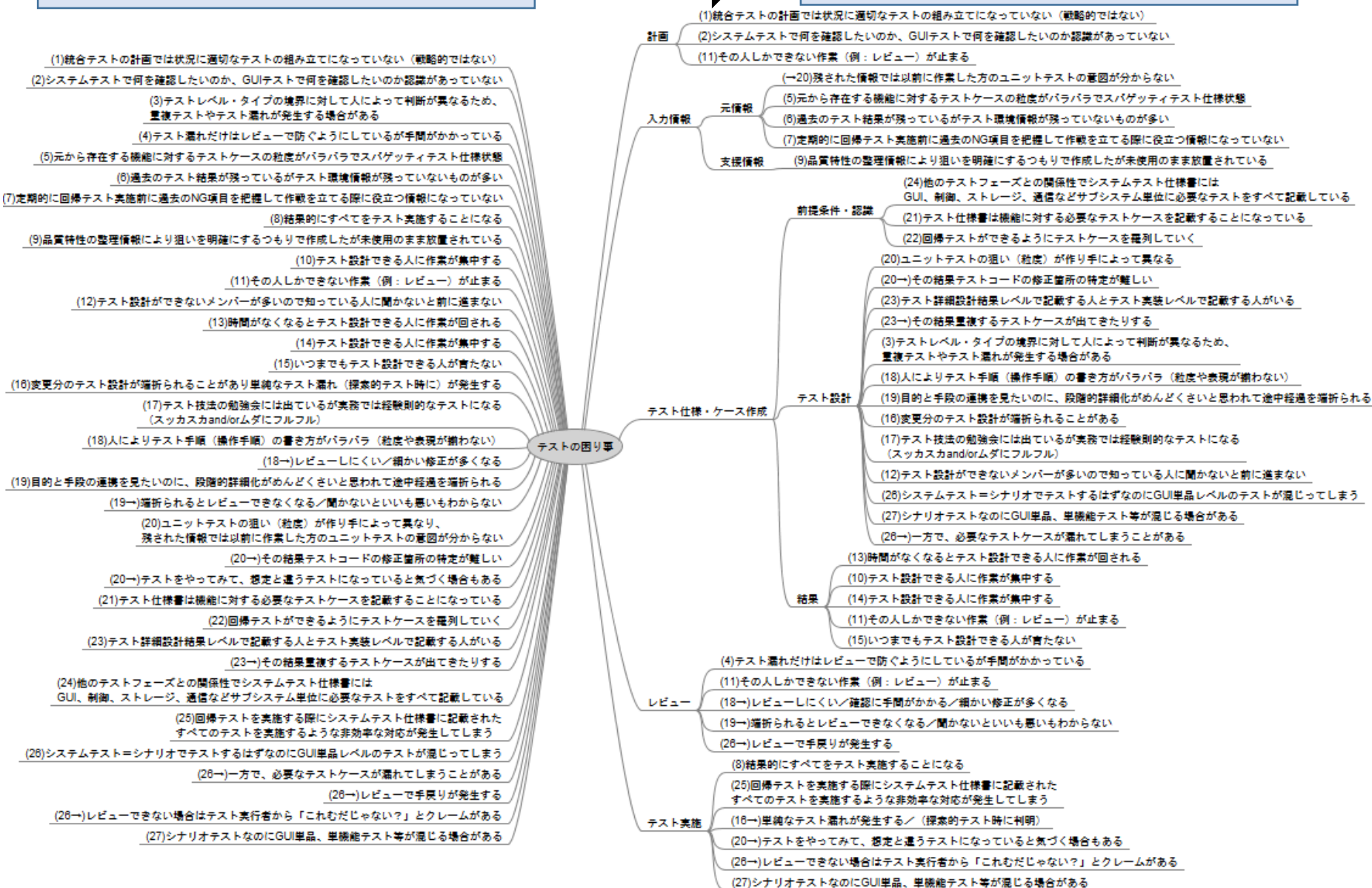
- 少ないことでどのような困り事が発生している？
- できる人に作業が集中する
- その人しかできない作業（例：レビュー）が止まる
- テスト設計でやることがわからない
- その結果どのような困り事が発生している？
- できない局面では誰かに聞きに行かないと進まない
- 時間がなくなるとできる人に作業が回される
- できる人に作業が集中する
- いつまでもできる人が育たない
- 変更分のテスト設計をやっていない
- どうして？
- やらなければならないと思っていない
- その結果どのような困り事が発生している？
- 小さい案件ほど単純なテスト漏れ（探索的テスト時に）が発生する
- テスト技法を使ったためしがない
- その結果どのような困り事が発生している？
- 技法の勉強会には出ているが実務では使っていない
- 結果的に経験則的なテストになる（スッカスカand/orムダにフルフル）
- テストケースの書き方があいまい
- 具体的にどのような書き方になっている？
- 人により手順（操作手順）の書き方がバラバラ（粒度や表現が揃わない）
- その結果どのような困り事が発生している？
- レビューしにくい／細かい修正が多くなる
- テスト設計からテスト仕様書に落とし込むプロセスがあいまい
- 具体的にどのようなプロセスになっている？
- 目的と手段の連携を見たいのに、段階的詳細化がめんどくさいと思われる
- その結果どのような困り事が発生している？
- 端折られるとレビューできなくなる／聞かないといいも悪いもわからない
- ユニットテストの狙い（粒度）が作り手によってことなる
- ユニットテストの狙いの事例（どのように異なる内容なのか？）
- 残された情報では以前に作業した方のユニットテストの意図が分からない
- その結果どのような困り事が発生している？
- テストコードの修正箇所の特定が難しい
- テストをやってみて、想定と違うテストになっていると気づく場合もある
- テスト仕様書の役割があいまい
- 具体的にどのような状態をあいまいと言っている？
- 機能に対する必要なテストケースを記載することになっている
- 回帰テストができるようにテストケースを羅列していく
- テスト詳細設計結果レベルで記載する人とテスト実装レベルで記載する人がいる
- その結果どのような困り事が発生している？
- 重複するテストケースが出てきたりする
- UI単体のテストは分冊したい
- どうして分冊したいの？分冊していないとどのように困るの？
- 他のテストフェーズとの関係性でシステムテスト仕様書には
- GUI、制御、ストレージ、通信などサブシステム単位に必要なテストをすべて記載している
- 回帰テストを実施する際にシステムテスト仕様書に記載された
- すべてのテストを実施するような非効率な対応が発生してしまう
- UI単体のテストはどれぐらいの粒度のテスト
- UI単体のテストと統合確認のシナリオテストの境界がわからない

問題意識の背景＝リアルなQCD問題関連事象や困り事を掘り下げて把握する→改善目標設定に役立つ

実存するQCD問題関連事象・困り事の列挙と分類

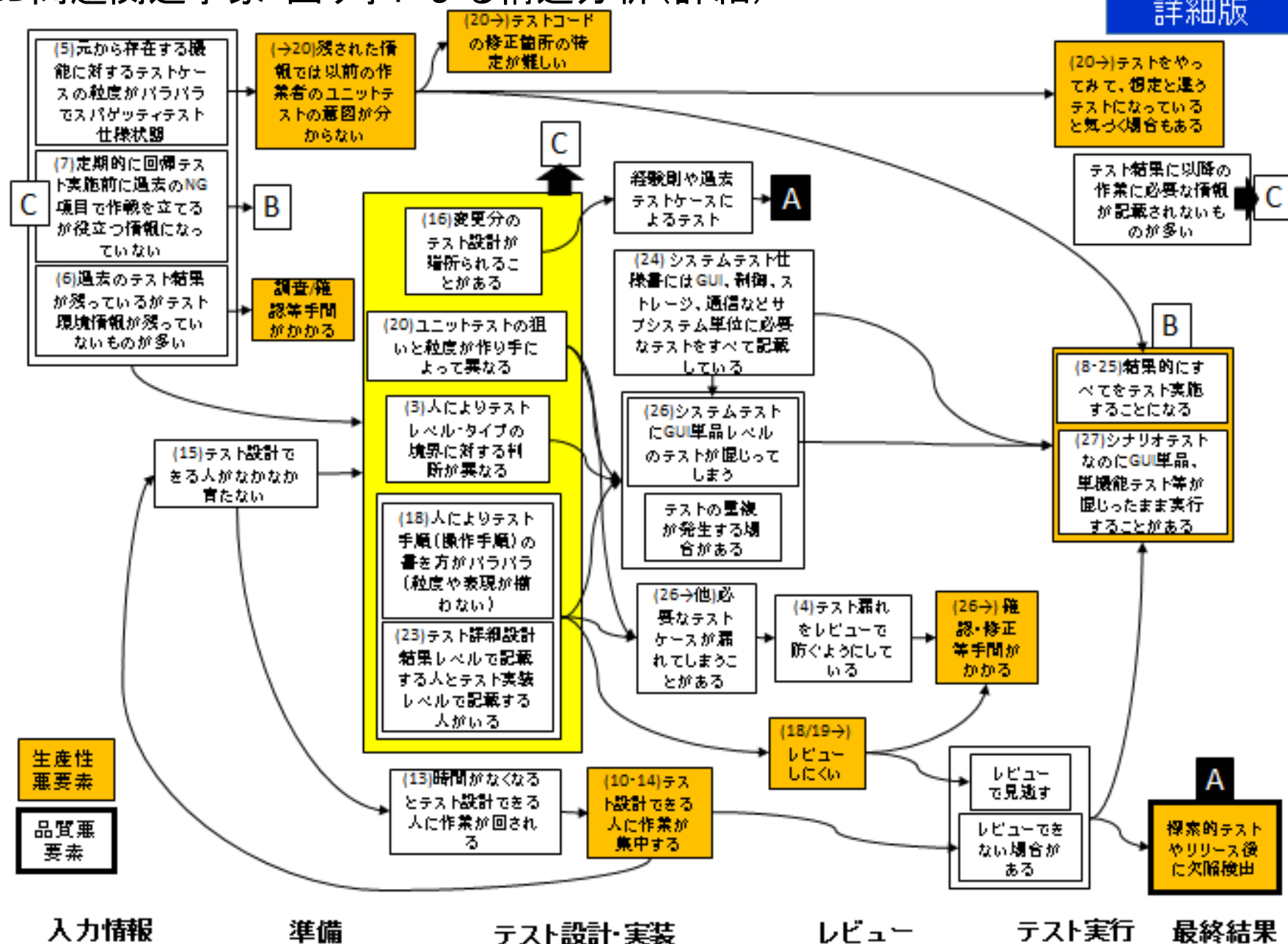
実際のふりかえり結果

整理・分類結果



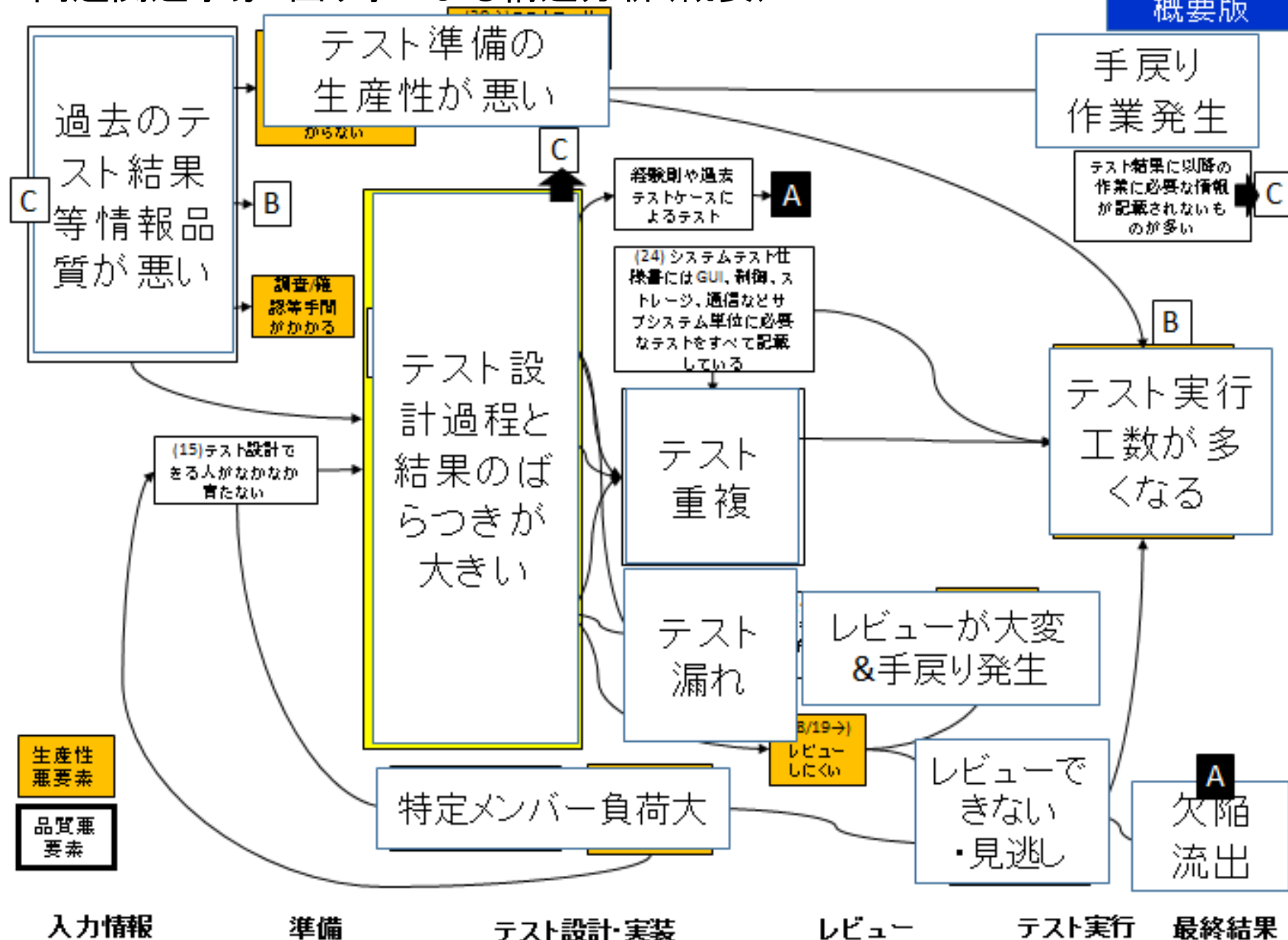
QCD問題関連事象・困り事による構造分析(詳細)

詳細版



QCD問題関連事象・困り事による構造分析(概要)

概要版



QCD問題関連事象・困り事から アセスメント結果を見直してみると...

		初期レベル	コントロールレベル				効率化レベル				最適化レベル		
1	利害関係者のコミットメント		1	2	3	4	1	2	3		1	2	3
2	関与の度合い		1	2	3	4	1	2	3		1	2	
3	テスト戦略		1	2	3	4	1	2	3		1	2	
4	テスト組織		1	2	3	4	1	2	3	4	1	2	3
5	コミュニケーション		1	2	3	4	1	2	3		1	2	
6	報告		1	2	3	4	1	2	3		1	2	
7	テストプロセス管理		1	2	3	4	1	2	3		1	2	
8	見積もりと計画		1	2	3	4	1	2	3	4	1	2	3
9	テストリソース		1	2	3	4	1	2	3	4	1	2	
10	欠陥管理		1	2	3	4	1	2	3	4	1	2	3
11	テストウェア管理		1	2	3	4	1	2	3		1	2	3
12	手法の実践		1	2	3	4	1	2	3	4	1	2	
13	テスト担当者のプロ意識		1	2	3	4	1	2	3	4	1	2	3
14	テストケース設計		1	2	3	4	1	2	3	4	1	2	3
15	テストツール		1	2	3	4	1	2	3	4	1	2	3
16	テスト環境		1	2	3	4	1	2	3	4	1	2	3

見直し
対象領域

修正

修正

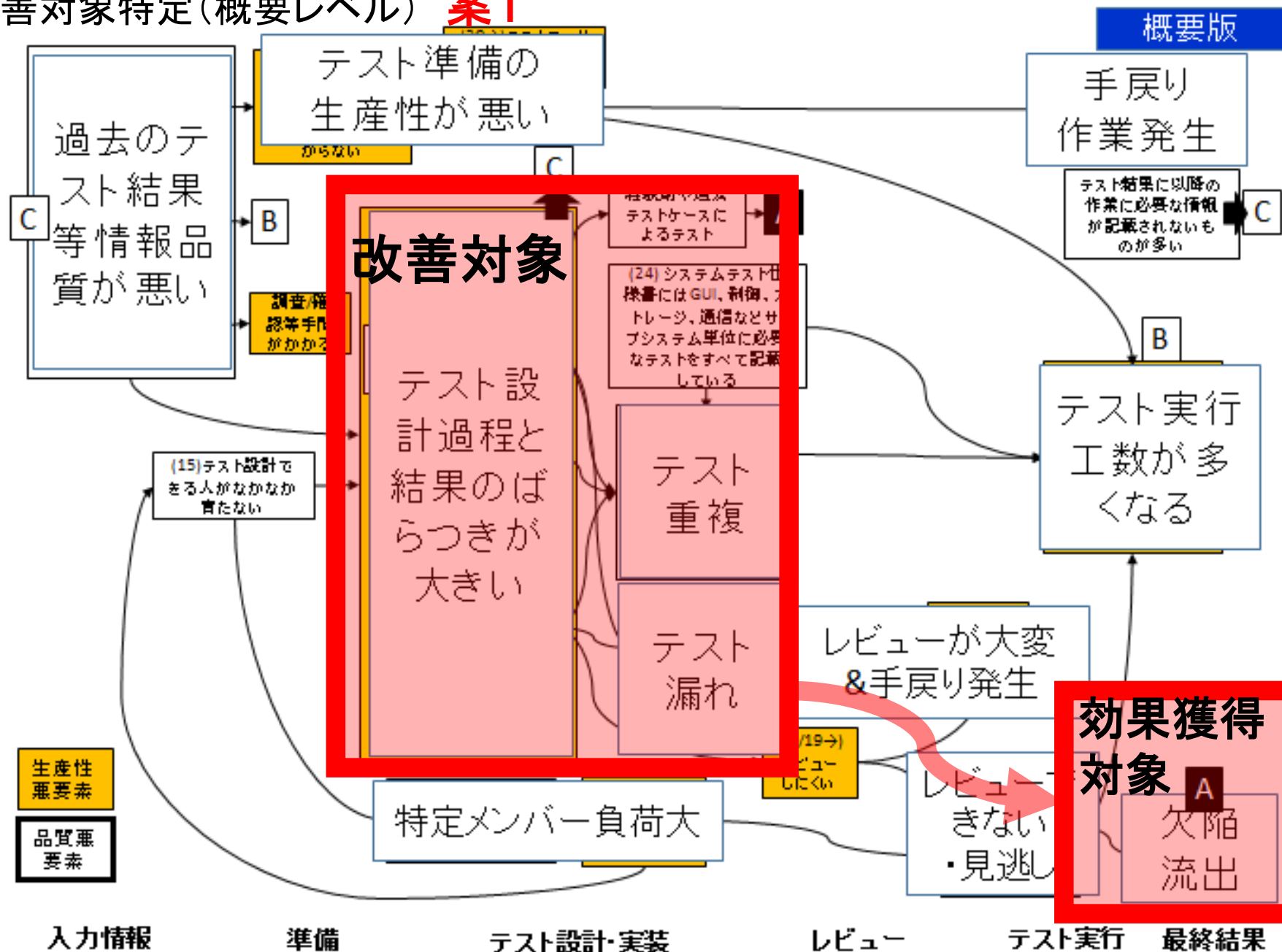
修正

修正



現地調査の結果から地図を書き換えよう

改善対象特定(概要レベル) 案1



改善対象特定(概要レベル) 案2

概要版

改善対象

過去のテスト結果等情報品質が悪い

テスト準備の生産性が悪い

手戻り作業発生

テスト結果に以降の作業に必要な情報が記載されないものが多い

経験則や過去のテストケースによるテスト

(24) システムテスト仕様書には GUI、制御、ストレージ、通信などサブシステム単位に必要なテストをすべて記載している

テスト重複

テスト設計過程と結果のばらつきが大きい

(15) テスト設計できる人がなかなか育たない

効果獲得対象
テスト実行工数が多くなる

発生頻度や欲しい効果などから案1を採用することに

手戻りが大変発生

レビューできない・見逃し

欠陥流出

生産性悪要素

品質悪要素

入力情報

準備

テスト設計・実装

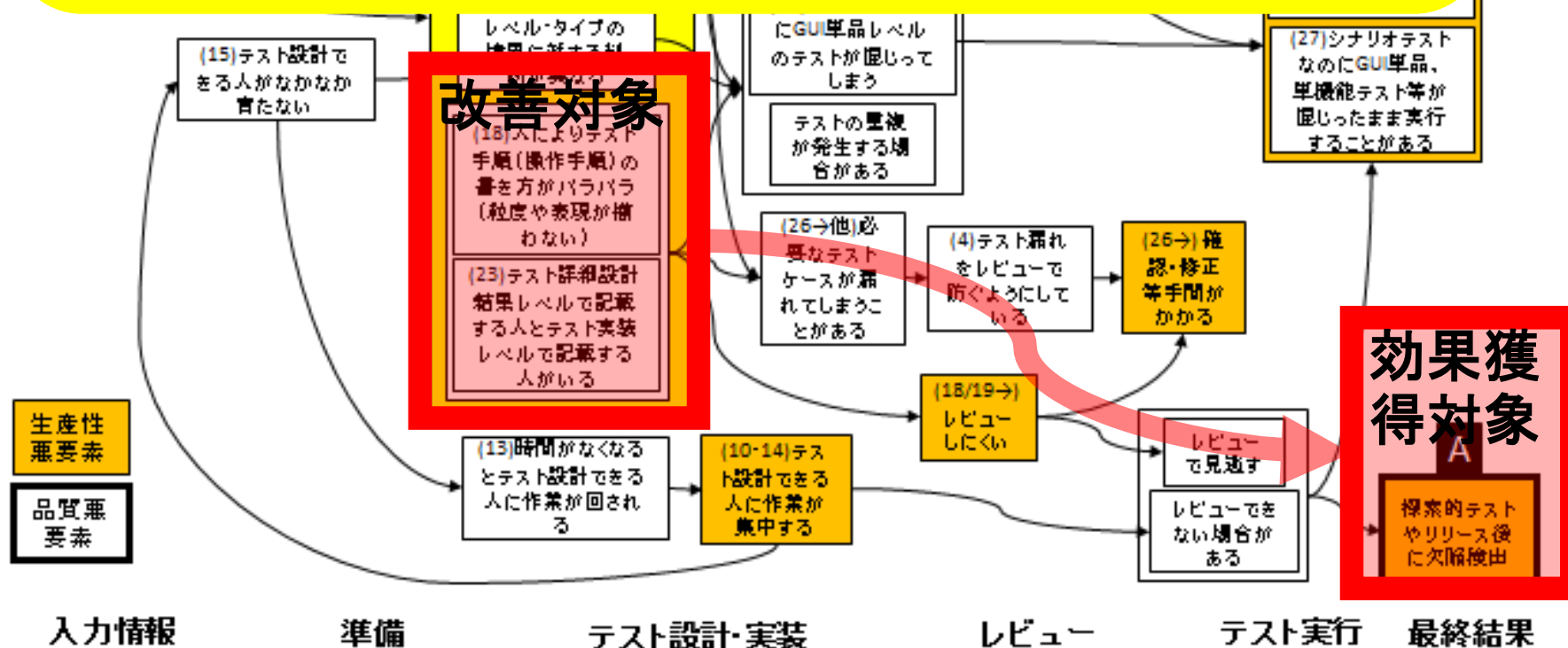
レビュー

テスト実行

最終結果

案1をベースに詳細レベルで改善対象を掘り下げ、さらに対象を絞り込む

→改善規模・工数・期間が小さくなり、短期間で結果が把握できる&効果を実感しやすくなる／失敗した場合の被害も最小に



改善施策要件候補洗い出し(TPI Next Practices)

11. テストウェア管理

コントロールレベル

- 1 テストベースとテスト対象およびすべてのテストウェアを、名前とバージョンで特定している。
- 2 各テストケースを、テストベース文書に明白な方法で関連付けている。
- 3 テストチームは、テストウェアの管理下のすべてのアイテムにアクセスできる。
- ④ テストウェア、テストベース、テスト対象の扱い方が明確に規定され、テストチームに伝えられている。

効率化レベル

- 1 テストベースとテスト対象およびすべてのテストウェアを、名前とバージョンで参照できる。
- 2 テストケースと要件のトレーサビリティが確保されている。
- ③ テストウェア管理は、論理的な補完構造と、役割および権限の構造によって支えられている。

最適化レベル

- 1 テストプロジェクト終了後にどのテストウェアを保持するかをテストプロジェクト開始時に合意し、テストプロジェクト実施中に見直している。
- 2 再利用に備えたテストウェア保持に関するガイドラインが入手可能な状態にあり、テストウェアの再利用を測定している。
- 3 プロジェクト終了時に保守に引き渡すテストウェアが、保守されないテストウェアと容易に分離できる。

12. 手法の実践

コントロールレベル

- ① テストプロセスは、明文化されたテスト手法に従っている。テスト手法には、一連のテスト活動、テストプロジェクトの成果となるテストプロダクト、作業の途中で発生する追加要求について記述している。
- 2 テスト手法は、プロジェクトが用いている開発手法に適合している。
- 3 テストプロジェクトにとって、実装したテスト手法が実用的なものであると認識されている。

効率化レベル

- ① テスト手法には、すべてのテスト活動に関するゴールと役割、および用いるべき技法と事前条件が明文化されている。
- ② 完全かつ包括的なテンプレートの一式が、テスト手法の一環として提供されている。
- 3 テスト手法の各要素について、必須／条件付き／任意のいずれかが記載されている。
- 4 必須と条件付きの要素について、実践事例がある。

最適化レベル

- 1 テストチームは、テスト手法について組織的にフィードバックを行っている。
- 2 実装したテスト手法を継続的に強化し、改善している。

14. テストケース設計

コントロールレベル

- ① テストケースを論理レベルで記録している。
テストケースには、以下の説明項目を含む。
- ② a) 開始時の状況
b) 変更プロセス=実施するテストアクション
c) 予測される結果
- ③ テストケースにシステムの詳細な振る舞いを記述することで、テストベースのどの箇所がテストの対象であるかが把握できる。

効率化レベル

- ① テストケースは、テスト組織の同僚が見ても理解でき、保守できるものになっている。
- 2 テストケースによって達成できるテストベースのカバレッジレベルが明確である。
- ③ テストケース設計に正式なテスト設計技法を用いている。
- 4 テストケースが設計できないような品質特性のテスト作業には、チェックリストを用いている。

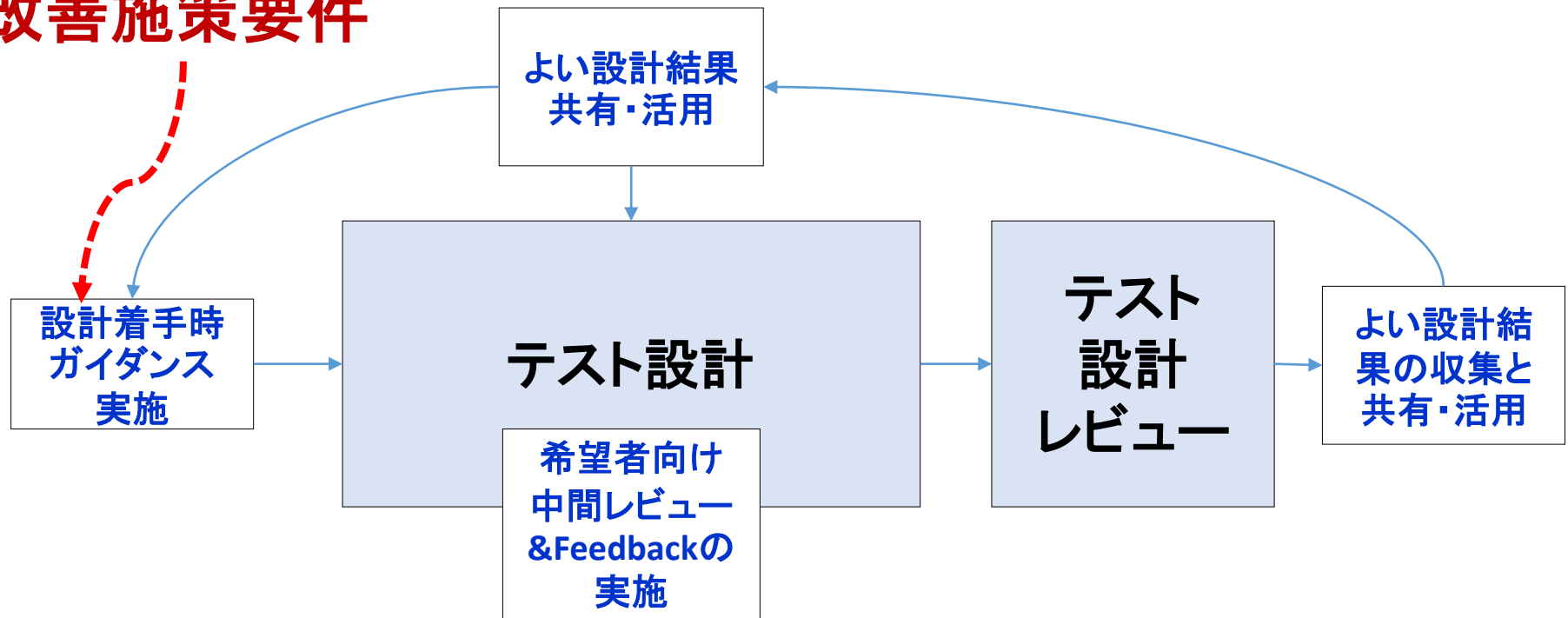
最適化レベル

- 1 次のフェーズ(次のテストレベルや本番)で発生した欠陥を分析し、テストケースの正確性や有効性の向上につなげている。
- 2 テストケースそれぞれの妥当性と保守性についてチェックし、評価している。
- 3 テスト設計技法を、将来さらに再利用するために評価し、調整している。

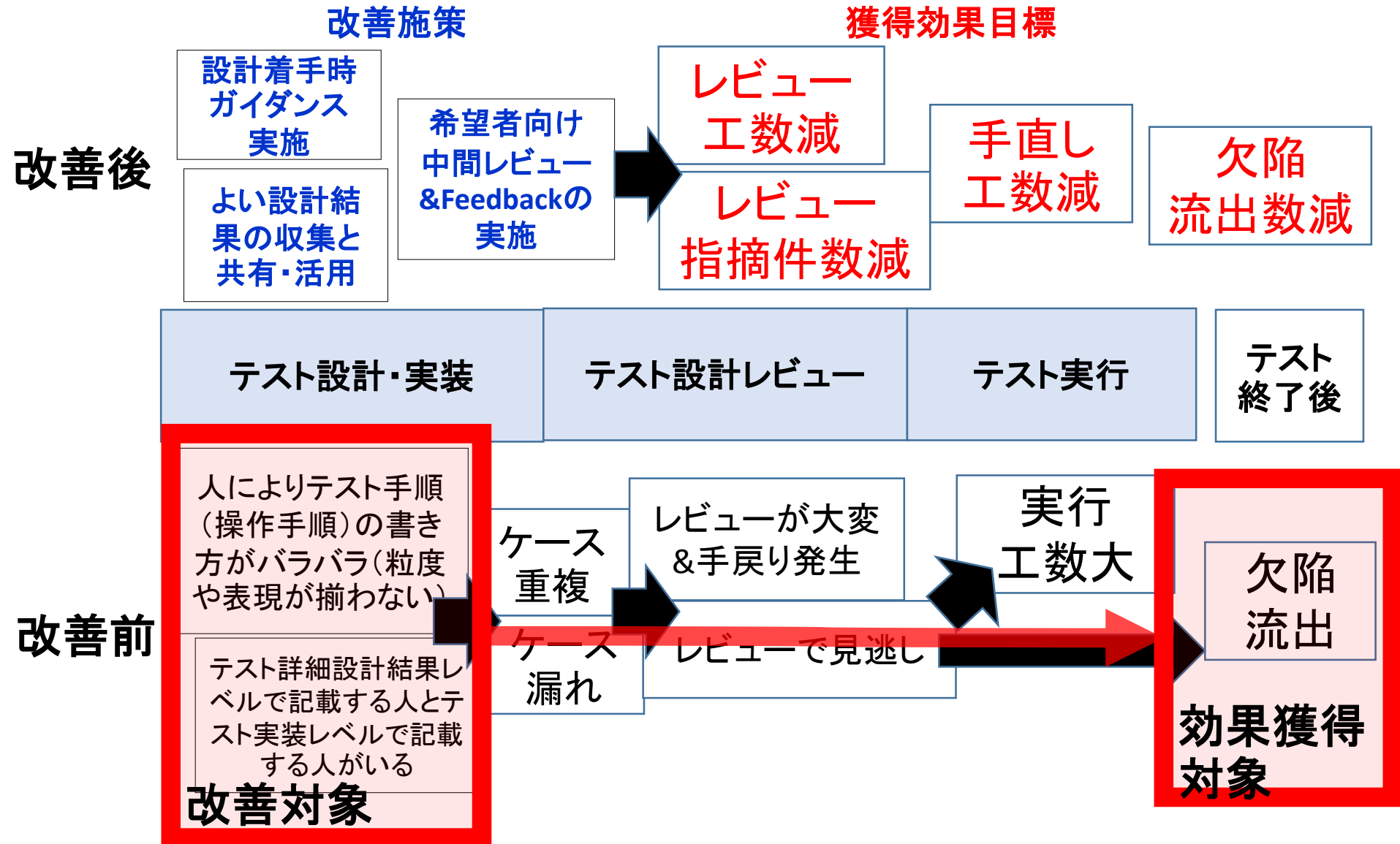
プロセスモデルのプラクティスは、ものづくりへの効果と効率を高めるための施策要件の集合体

改善施策要件を参考に 改善施策を明確化

改善施策要件



改善施策と獲得効果目標の設定



当アプローチのSTEP全体像(TPI Next活用箇所と利点・効果)

セルフアセス終了後の実施STEP	TPI Next活用	利点や効果
メンバーふりかえりによるテストへの問題意識の共有		メンバーの問題意識から始めると当事者意識が高まる+改善効果を実感しやすい
テストプロセス全体像と要素関係性の把握	テストプロセス全体像モデル(TP全体像M)	全プラクティスでの関係性把握は困難／プロセスエリアレベルなら把握しやすい
関係者の問題意識分析	TP全体像Mにメンバーの問題意識を割り付け集中箇所を特定	問題意識が集中しているところに着眼すると関係者を巻き込みやすく、改善効果も高くなる
実在するQCDS上の問題点把握		通常「QCD問題事象の解決手段」を問題として誤認識していることが多い／背後に隠れているQCD問題事象を把握することで実感しやすい改善効果や目標を設定できる
困り事・問題点による構造分析		事象・事実レベルでQCD問題群を構造的に把握すると腹落ちしやすい
改善対象特定		まずは大枠で改善対象と目指す効果を把握(特定)すると考えやすい
改善対象絞込み		絞り込むことで改善リスクを最小化しつつ効果を実感しやすくする
改善施策要件候補洗い出し	改善施策要件洗い出し	プラクティスは汎用的→施策要件として活用するのが適切
具体的施策と獲得効果目標の設定		改善をシナリオ化し成功確率を高める

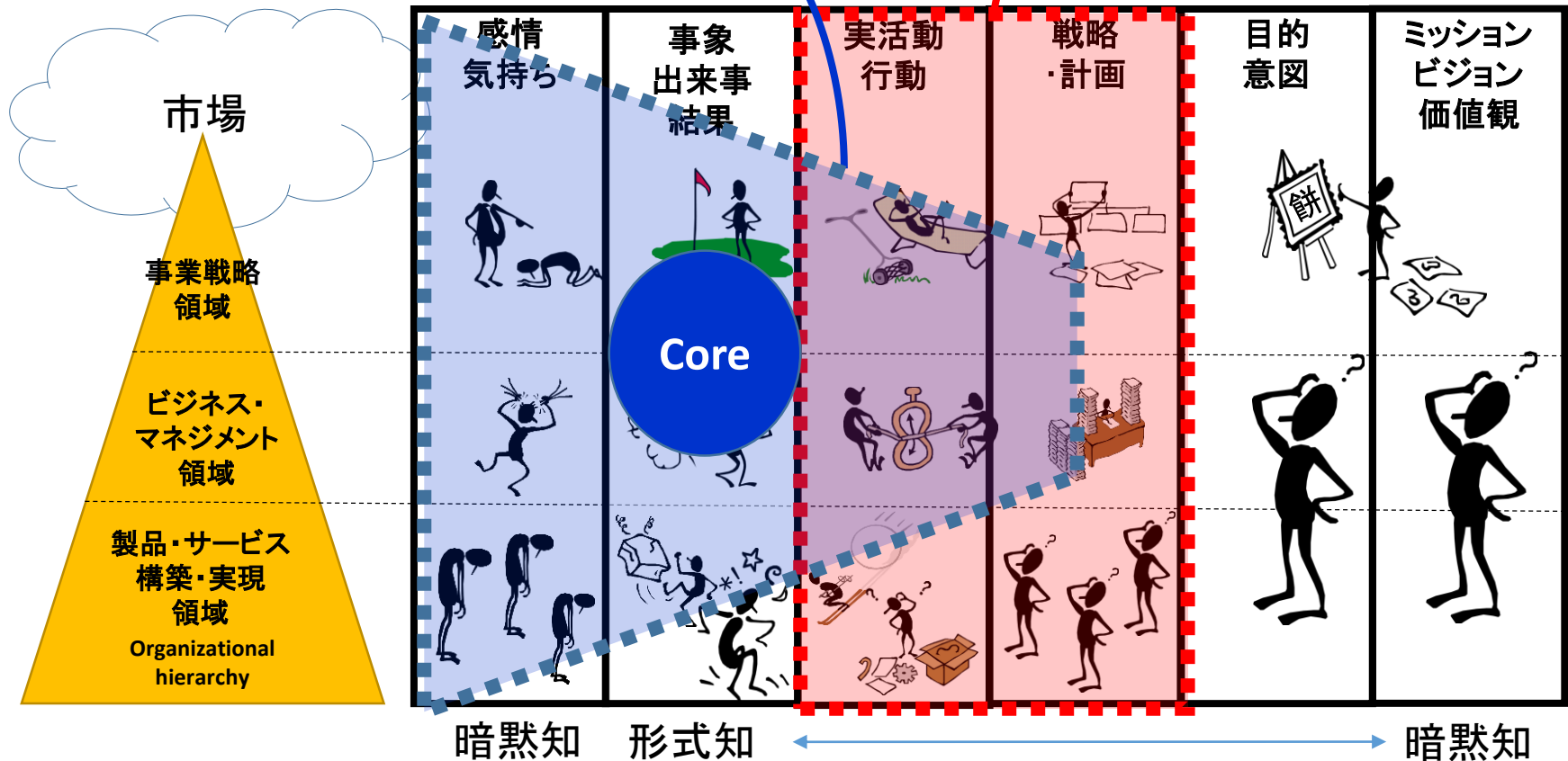
プロセスモデル適用の典型的失敗例との比較

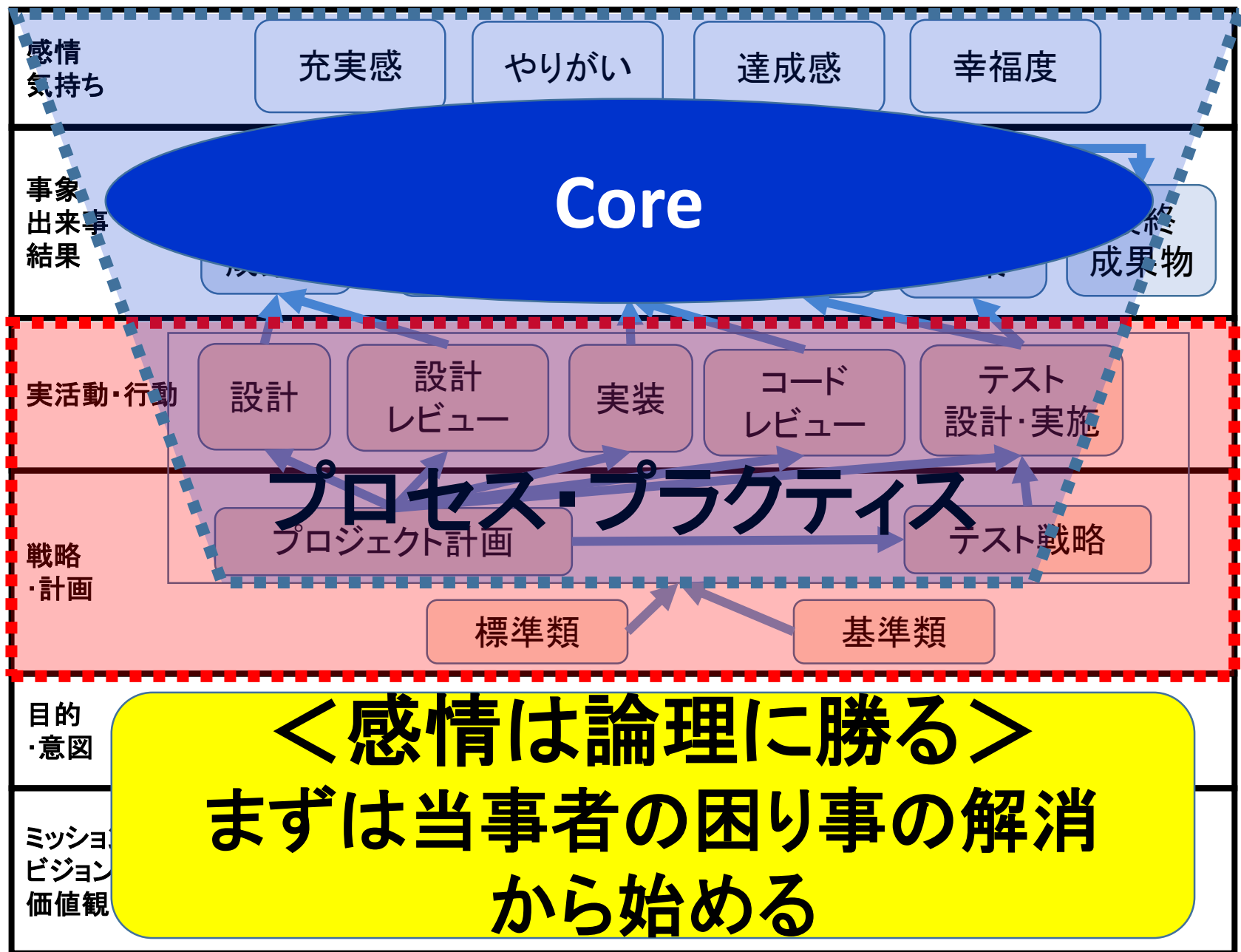
主な改善過程	プロセスモデル適用の典型的失敗例	今回のアプローチ
現状把握 現状分析	16キーエリア146プラクティスフルセル ファセスメント→工数大になるため代表 者(例:リーダー)が実施する傾向大＝実 態を反映しない結果になる可能性が高 い	TP全体像M上の集中箇所に関わる QCD問題関連事象・困り事を洗い出し、 構造分析して状況を明確にする／ふり かえり結果などメンバーが感じている 問題意識から始められる
改善対象特定	アセスメント結果から有効な改善対象 を特定するのは有識者でなければ困 難→代表者の一声で決める、またはモ デル推奨に従うが、どちらもうまくい かないケースが多い	QCD問題関連事項・困り事の構造分析 結果から改善対象を概要レベルで特 定し、さらに詳細レベルで把握する／メ ンバーの問題意識をベースにした改善 対象が特定できる
改善目標	アセスメント結果から有効な改善目標 を設定するのは簡単ではない→モデル 適合を目標に→プロセスが重くなる＋ 効果が実感しにくい	QCD問題関連事項・困り事の解消を目 標にできる／実務の困り事が減る効果 を実感しやすい
改善施策	プラクティスを直接単品追加しがち→ 改善するほど重くなる	現状のプロセスをベースに改善要件を 満たす施策を展開できる
改善期間・効果 実感など	長期化しやすい・効果実感を得られに くい／失敗と分かるまでに時間がかか る＆見直しも大がかりになる	短期間で効果を実感しやすい／失敗し た際の見直しも容易

TPI Nextと当アプローチの違い

当アプローチ
のSCOPE

TPI Next(プロセス
モデル)のSCOPE





当アプローチの特徴

- 利点

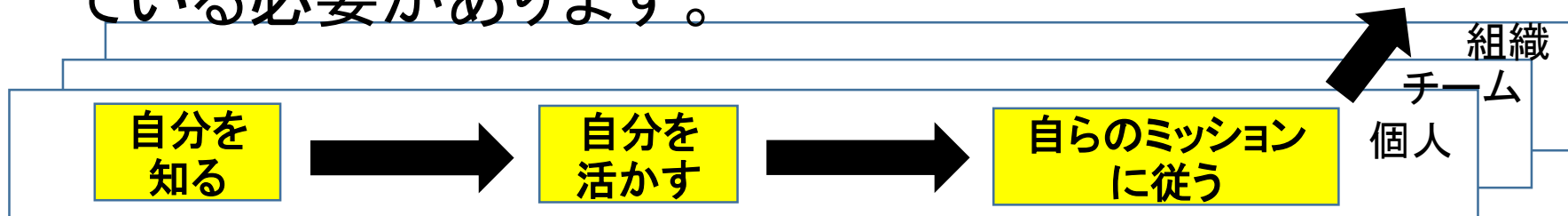
- 146プラクティスのフルアセスメントを行わずとも、16キーエリアの全体像にメンバーの問題意識をマッピングすれば改善対象領域を特定できる。(ふりかえりが実施されていればその情報から始められる)
- メンバーの問題意識をベースに進めることで、チーム全員の当事者意識が高められる&改善効果が実感しやすくなる→メンバーを巻き込んで改善が継続しやすくなる。
- プロセス改善に取り組みながらTPI Nextの理解度向上に応じて活用深度を変化させながら進められる。→より高度な改善活動に段階的にシフトする基盤になる。

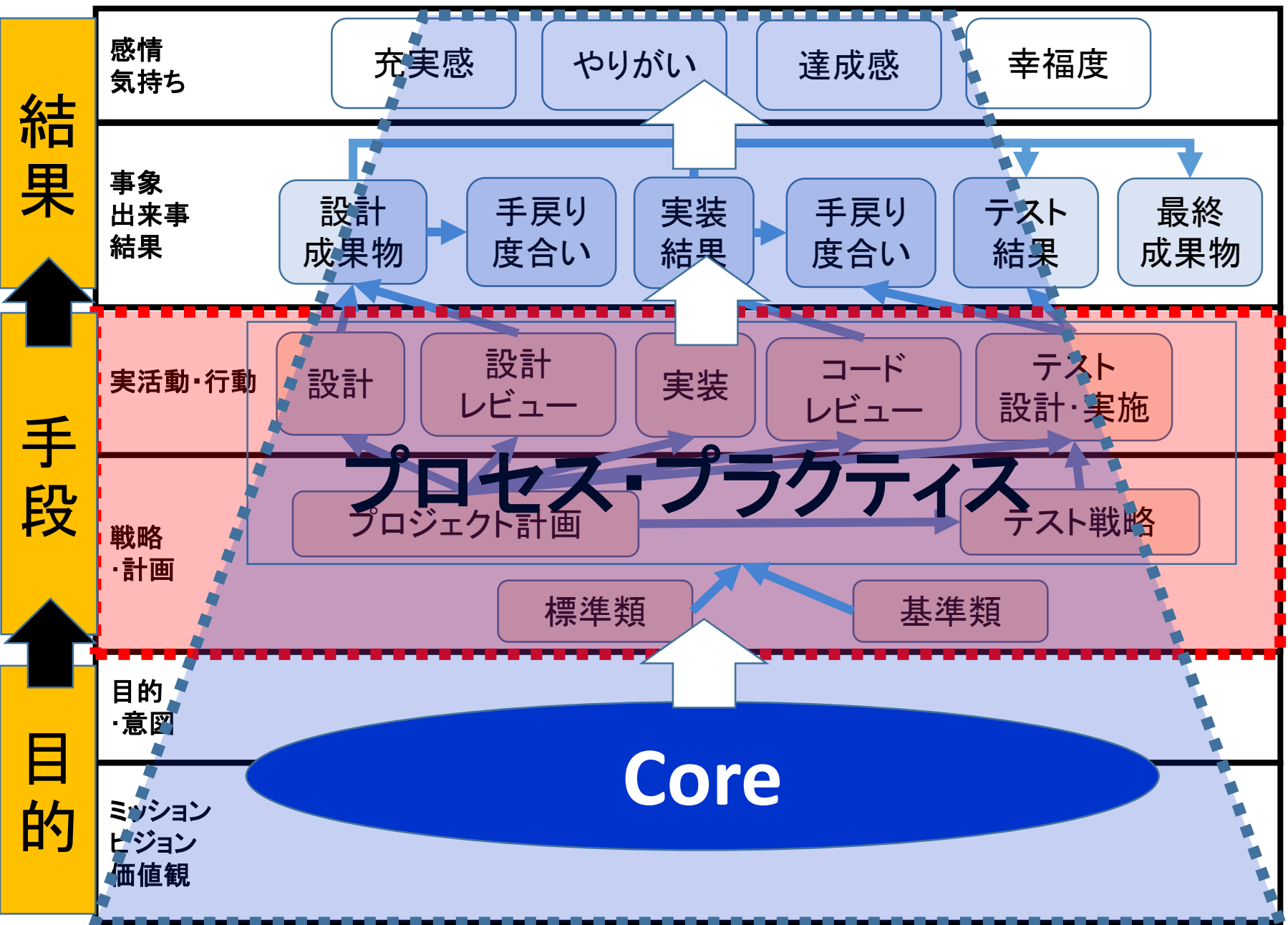
- 懸念点

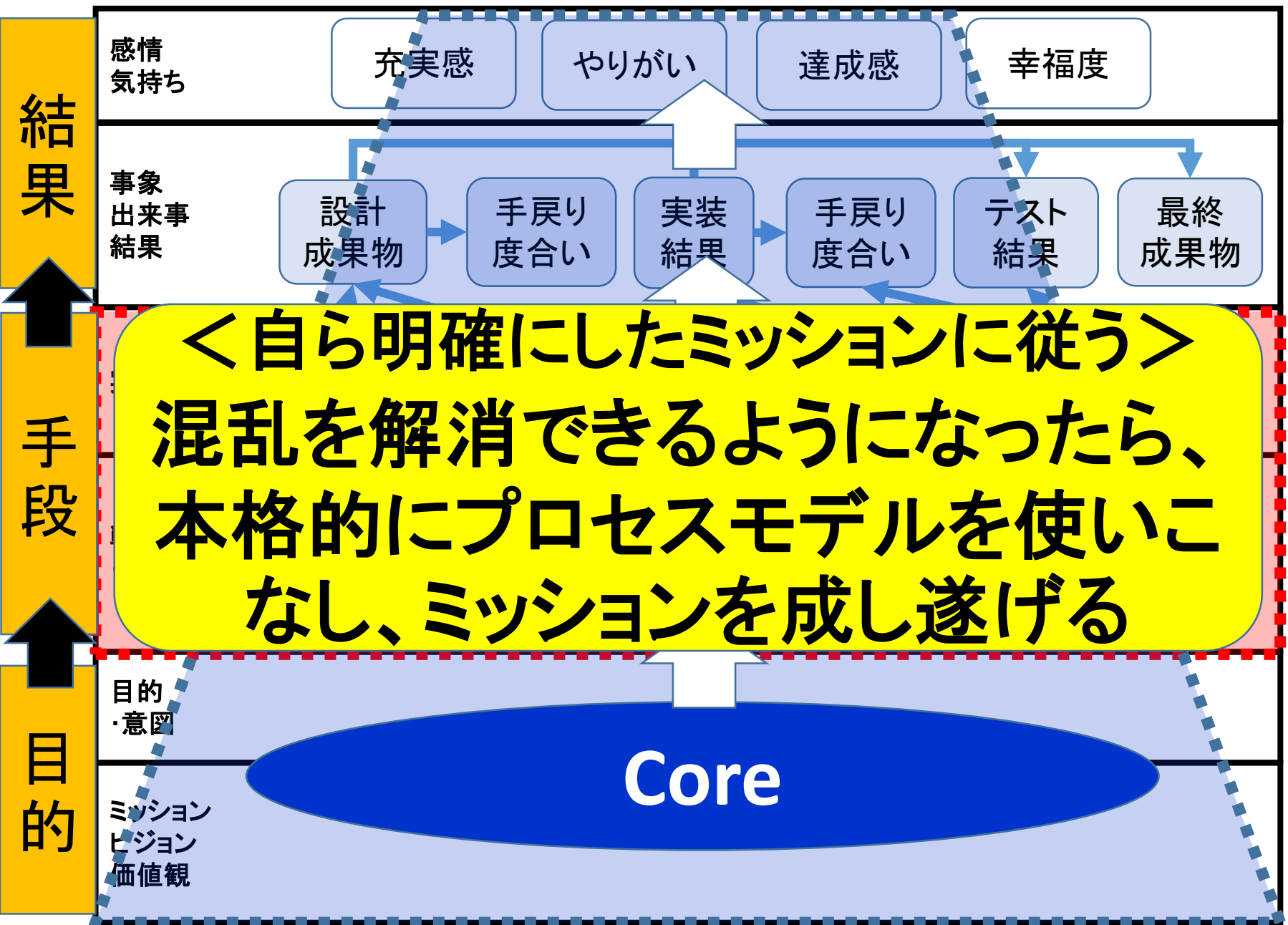
- 専門家がアセスメントを行わないため客観性や正確性が低くなる場合もある。
- QCDS問題関連事象・困り事の収集と構造分析実践スキルが必要。当初は有識者の支援を受けるなどの対策は必要。

そしてその先へ

- 当アプローチは、「自分たちの現在(いま)を知る」ことで実務メンバーの心に火を灯し、自ら走り始めるためのものです。
- 実際に走り始めて徐々に「自らを活かす」ことができるようになったら、その次は「自らのミッションを明確にし、それに従う」ことにチャレンジする必要があります。
- その壁を越えるためには、ミッションや目的達成のためにプロセスモデルを有効活用できるようになっている必要があります。







参考文献

- TPI NEXT アセスメントツール日本語版
<http://www.tmap.net/system/files/TPI%20XLS%20versiev2.xls>
- JaSST'17北海道 JaSSTセッション
中山さん、テストプロセスアセスメントやってみたってよ
<http://jasst.jp/symposium/jasst17hokkaido/pdf/S6-1.pdf>
- 「ソフトウェアプロセス改善手法 SaPID入門」
日科技連出版社
<http://www.juse-p.co.jp/cgi-bin/html.pl5?i=ISBN978-4-8171-9510-4>
SaPID: <https://www.software-quasol.com/sapid2-0/>
- ソフトウェア・シンポジウム 2013 in 岐阜 (SS2013)
プロセスアセスメント結果の現実的・効果的活用方法の提案
<http://bit.ly/2Fx4IPq> <http://bit.ly/2Dgfx6l>
- Software Testing ManiaX Vol.10寄稿記事
ソフトウェアテストプロセス評価モデル微考w
<https://goo.gl/YbgmzG>