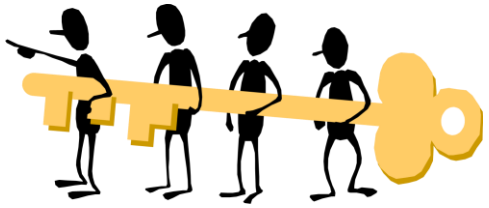


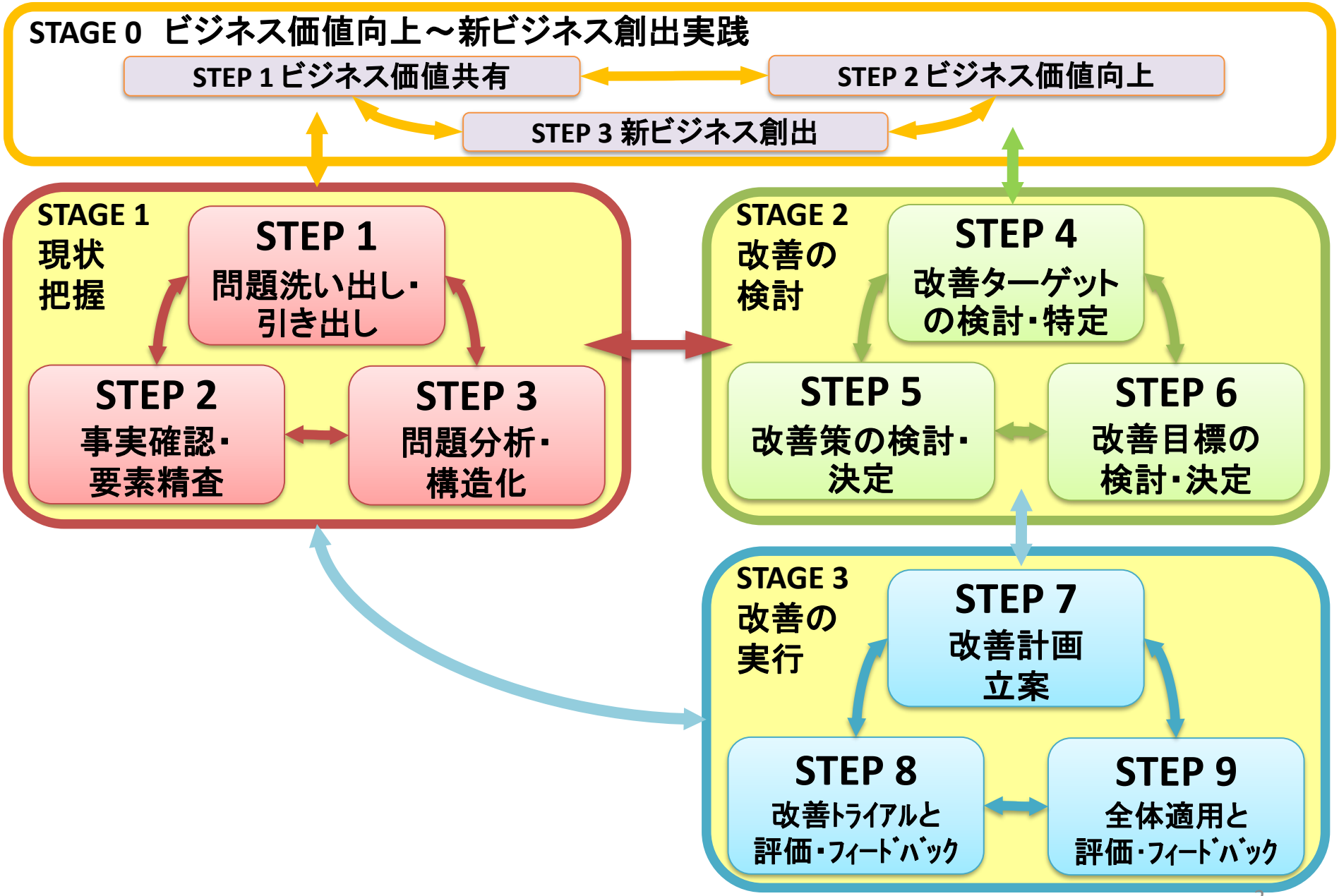
SaPID（サピッド）の概要

株式会社 HBA　Software Quasol 安達 賢二
adachi@hba.co.jp
http://www.software-quasol.com/

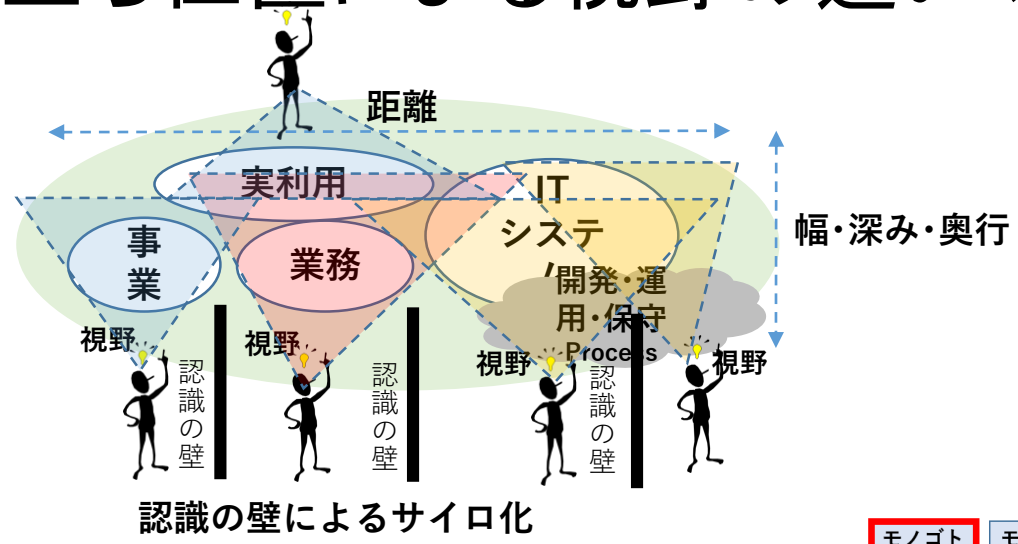
P.000 ←書籍「ソフトウェアプロセス改善手法SaPID入門」の記載頁数を指します。

※ “SaPID”は、株式会社HBAの日本における登録商標です。当スライドではTM,®などの表記を省略します。

SaPID Process Model



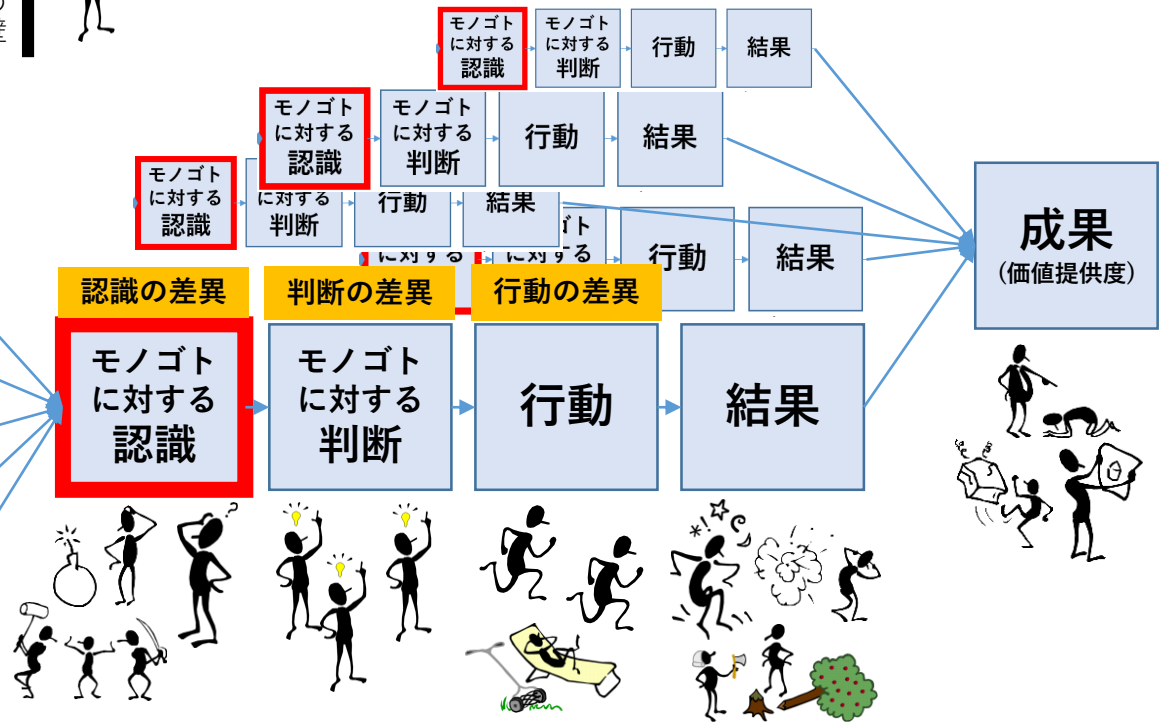
立ち位置による視野の違い等による認識不一致



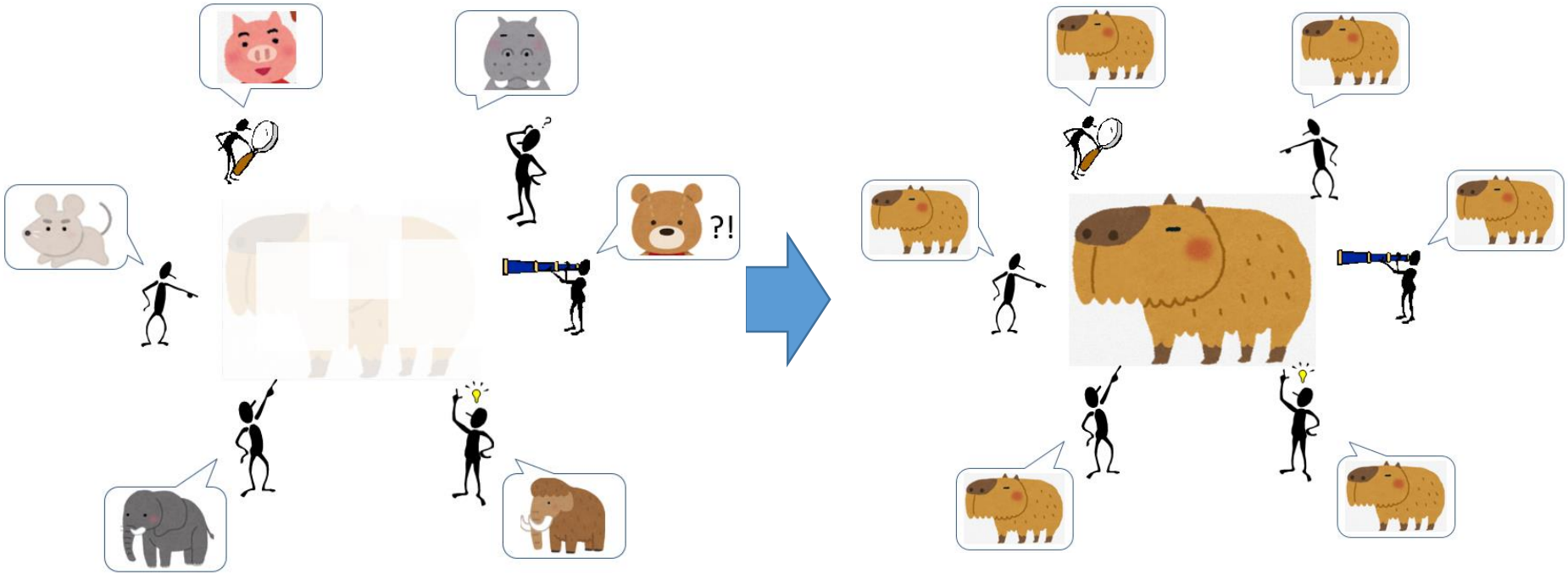
認識の壁によるサイロ化

この状況を打開するためにSaPIDを実践する

- 立ち位置で視野と見え方が異なる
- 立ち位置によって関心事が異なる
- システム・ソフトウェアは目に見えない
- 人は見たいものしか見ない
- モノゴトの表現・共有方法の自由度が高い



システム思考×デザイン思考で
全員の状況認識や目的をすり合わせる



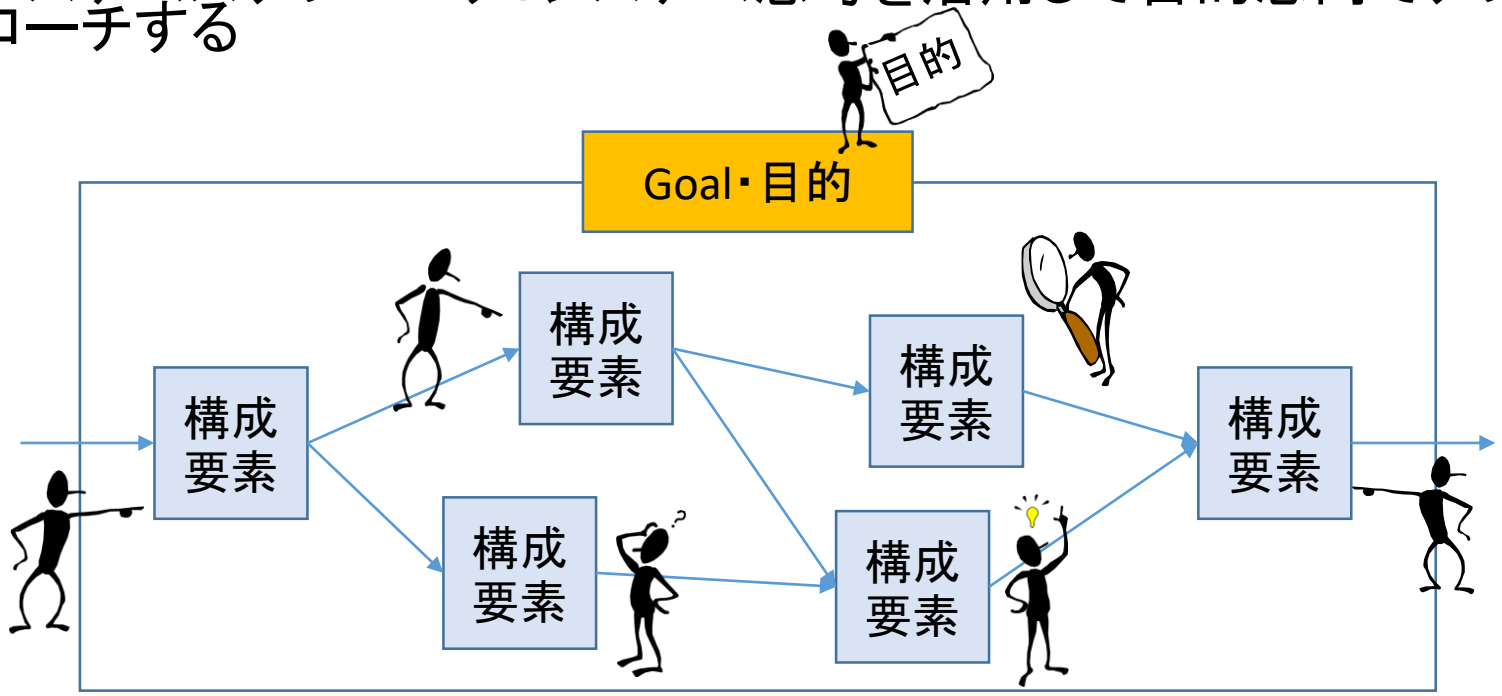
それぞれの立ち位置で見えるものが違う
そのまますり合わせないと・・・
判断を誤りやすい･意見がかみ合わない･
失敗が増える･言われたことしかしない

それぞれの立ち位置で見えるものを合わせると全体像と個別詳細が把握できる
状況把握が的確に･合意形成が容易に･
判断が揃いやすい･実践の動機づけになる

システム思考＝目的志向・指向

システムとは？
目的を達成するために、1つまはた複数の構成要素が相互に連携・作用するしくみ全体のこと

- システム思考：対象をシステムとみなして明確化する＝Systemic（俯瞰的）×Systematic（系統的）に思考する（ロジカルシンキングを含む）
- システムズアプローチ：システム思考を活用して目的志向でアプローチする



デザイン思考＝みんなで創る

【5つのMode】

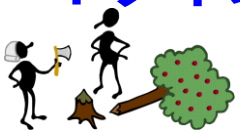
- Empathize：**共感**
- Define：**問題定義**

問題のリフレーミング

- Ideate：**創造**

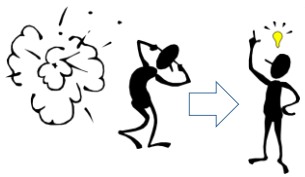
発散⇔収束

- Prototype：**プロトタイプ**



- Test：**テスト**

やってみて学ぼう！



引用元：-the d.school bootcamp bootleg

【4つのMindset】

- Human-Centered：**人間中心**

人間を意識して考える

- Collaborative：**協力的**

多様性を活かす

- Optimistic：**楽観的**

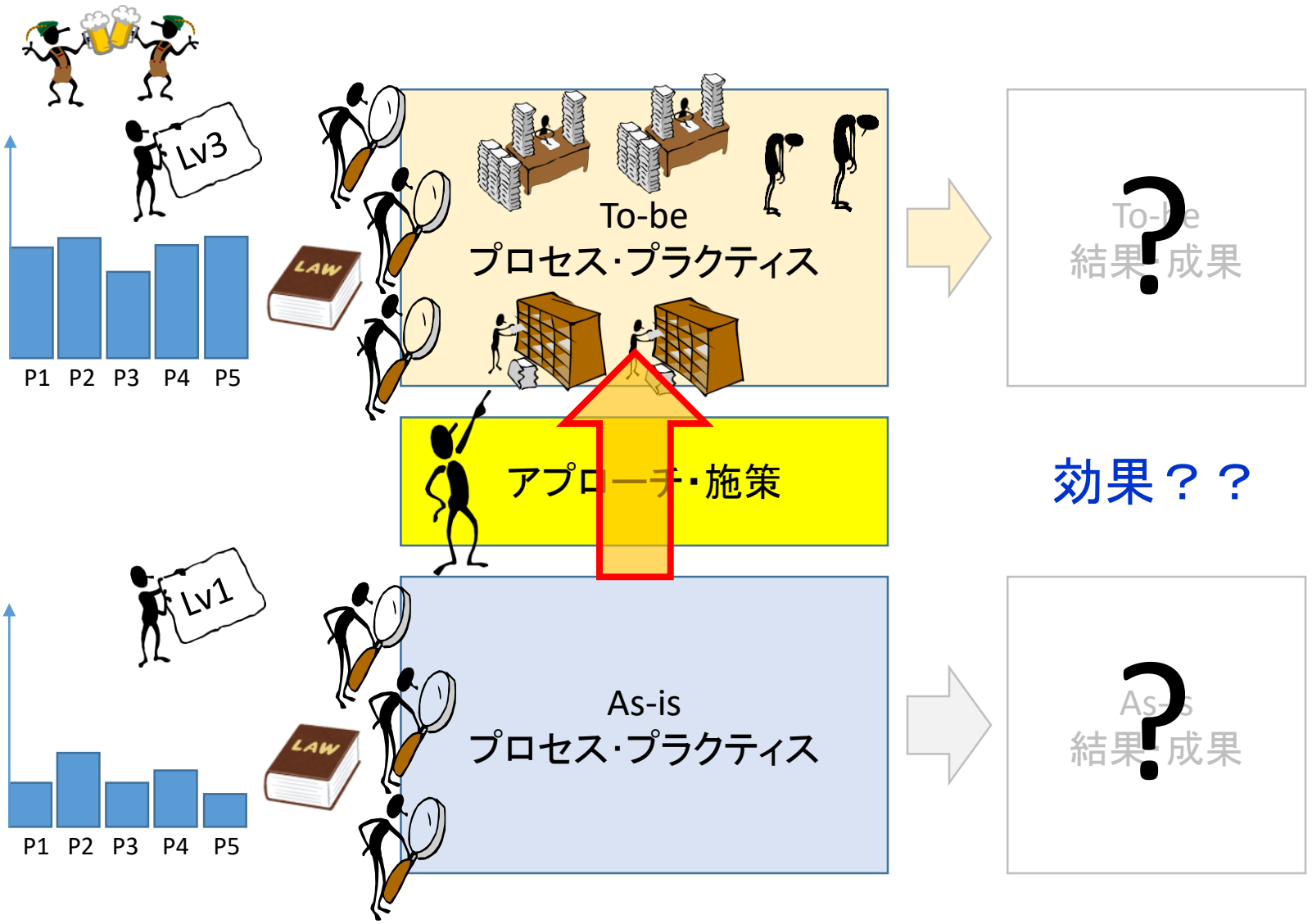
われわれにもできる！信念

- Experimental：**実験的**

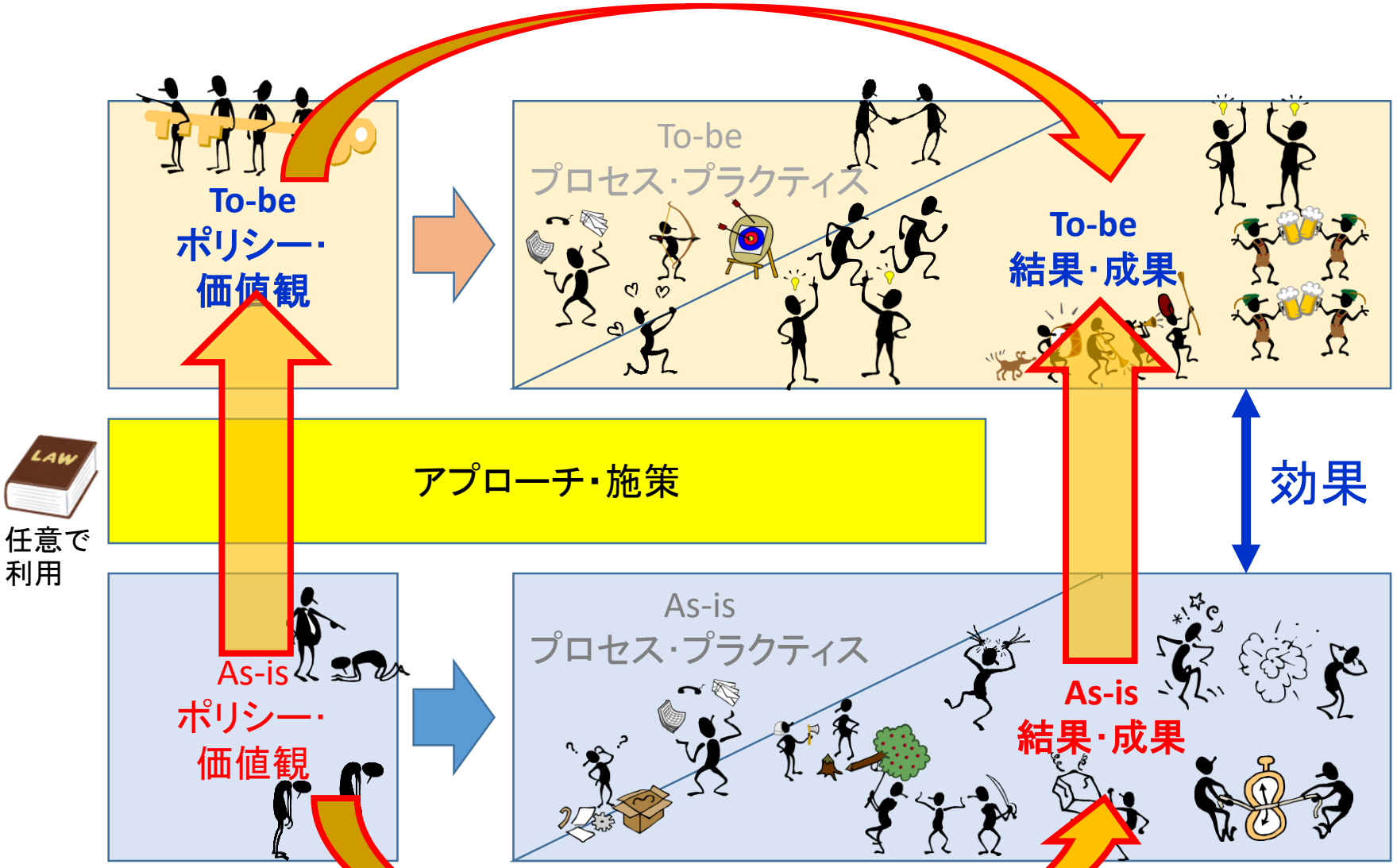
たくさんの試行経験から学ぶ

引用元：Design Thinking for Educators Toolkit,IDEO,2011

失敗するプロセスモデルベース改善のイメージ



SaPIDのアプローチイメージ



テーマ設定＝SCOPE設定

- テーマ設定により、把握する、検討する対象領域をやんわり決めます。

  →対象領域を決めることで、対象領域の事項に検討や議論を集中し、対象外事項を取り扱わないようにすることができる。

- テーマ設定の例
  - ○○開発プロジェクトの問題点
  - △△チームのパフォーマンス向上に向けて

STEP0:ビジネスモデルの明確化

これらはシステム思考の「目的」に当たります

①自らの仕事(業務)と役割

例:○○○○○サービス

②その顧客

直接顧客～間接顧客

この対応の過程で自らのミッションやポリシーは何か？を明らかになるのが理想です。

③サービスで顧客に提供している価値

顧客にとっての価値～自分にとっての価値

④自らの仕事・役割の評価方法

どうなれば成功？　何で評価する？

実際には、現状のポリシー／理想のポリシー、プロダクトアウト型／サービス提供型などに分かれます。この時点ではどちらになっても構いません。
SaPIDファシリテーターはこの時点でどちらを掲げたのかを把握しておきます。

ボードに配置してワークSTART!!

上部に「目的」を設定

①〇〇〇〇
〇サービス

②顧客：
□□□□□
▽▽▽▽

③提供している価値
□□□

④評価方法：
□□□□□
△△△△△

①～④の内容についても随時見直し（見直し前の内容も残し）ながら進めます。

「目的」の下部は「機能」や機能間連携、作用を明らかにするエリア

STEP1：問題点洗い出し・引き出し

必要に応じてチェックリストを併用すると効果的

納品後に多くの障害が発生

毎日残業＆休日返上対応が多い

プロジェクトの収支が赤字

要求事項の決定が遅延する

顧客クレーム多発

特にプロジェクト後半に進捗遅延が常態化した

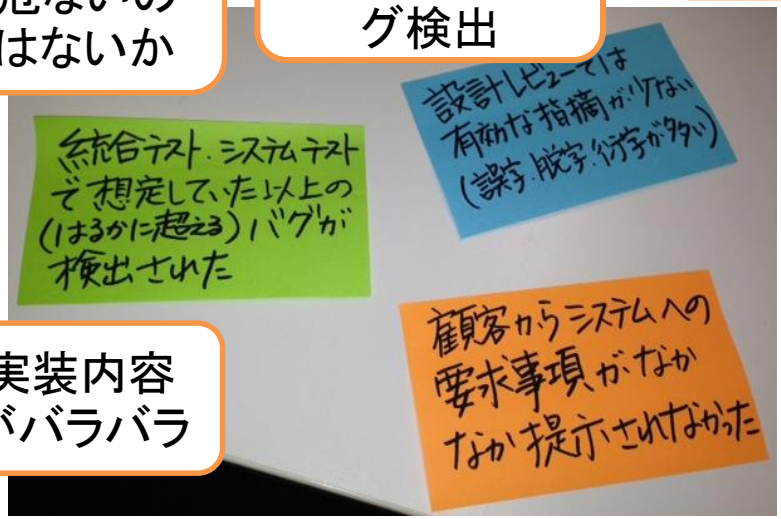
無責任な奴がいる

実装・テストは担当者の経験則に任せている

開発標準や各種判定基準がない

協力会社Aは危ないのではないか

システムテスト時に大量バグ検出



要求事項が何度も変更された

要求事項は記録されないことが多い

役割が長い間固定されている

実装内容がバラバラ

設計書はあったり、なかったりする

設計書レビューでは有効な欠陥指摘が少ない

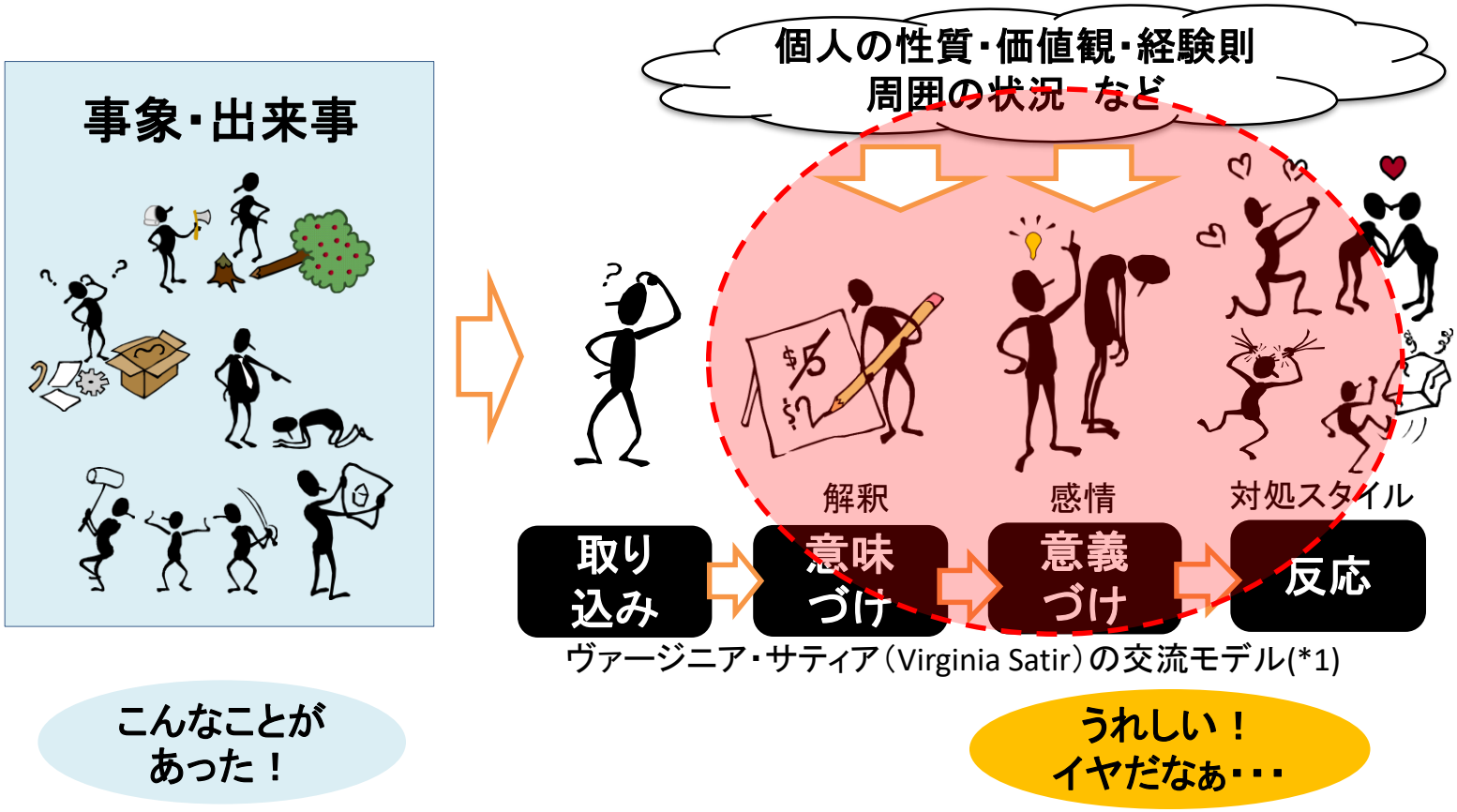
良い・悪いは抜きにして、何が起きているのか、どう感じているのか等をありのまま収集する

STEP1のポイント

デザイン思考

- テーマに対してそれぞれのメンバーが自分で感じている事項を素直に書いて出せるようにする。
  - 感情的に受けとめたモノゴトの背後には、インパクトがある事象や出来事が存在している、という仮説に基づいている。
  - 「テーマに対して」以外の制約を排除し、自由に書いてもらう
    - 問題点、困り事、あったらいいなと思うこと、これ嫌だなーってこと等
    - 変な制約があると間違いたくない意識が働いて書けなくなってしまう
    - 鬼軍曹や高圧的な上司などがいない場を作る等の場づくりが大事
- 付箋がなかなか出ない場合の対処例
  - 実務のライフサイクル（例：開発プロジェクトであれば、着手からリリース、安定稼働確認まで）の当初から終了タイミングまでの実務と想定される発生事象を段階的に思い浮かべてみる。
  - フェーズ単位に発生しそうな事象が列挙されたチェックリストで存在を確認してみる。

人間のモノゴトの取り込み～反応まで



ヴァージニア・サティア（Virginia Satir）の交流モデル(*1)

*1:参考　ソフトウェア文化を創る2 「ワインバーグのシステム洞察法」 共立出版　G.M.Weinberg

人が感情で反応する背景にはインパクトのある事象や出来事が存在する
～その反応の仕方を把握し、受けとめ、適切に変えていく＝成長

大切なのはできごとではない。できごとに対するわれわれの反応なのだ。

Gerald.M.Weinberg

技術リーダになるのは、彼らが失敗に反応するそのしかたによる。逆境を克服するだけでなくプラスに変える。（技術リーダとは）敗北を成功の跳躍台に使う能力を持った人々である。

STEP2：事実確認・要素精査

不適切な感覚・感情論なども手掛かりにして実在する問題・課題を具体的に捉える

- 納品後に多くの障害が発生
  - 15件
  - 218人時
- 毎日残業＆休日返上対応が多い
  - IT以降
  - 平均残業315h/m
- プロジェクトの収支が赤字
  - 対計画150%
- 要求事項の決定が遅延する
  - 対期日45日遅延
- 実装・テストは担当者の経験則に任せている
  - コード・テストケースのレビューなし
- 特にプロジェクト後半に進捗遅延が常態化
  - IT以降毎週10日以上の遅延が継続
- 顧客クレーム多発
  - クレーム8件
  - 4回客先説明
- ~~無責任な奴がいる~~
- 要求事項は記録されないことも多い
  - 存在15／対象28
- システムテスト時に大量バグ検出
  - 計画132件→実績208件
- ~~開発標準や各種判定基準がない~~
- 要求事項が何度も変更・追加された
  - 記録分のみ
  - 変更35・追加42
- ~~協力会社Aは危ないのではないか~~
- 実装内容がバラバラ
  - コーディング規約違反有が57%
- 設計書はあったり、なかったりする
  - 存在13／対象42
- 役割が長い間固定されている
  - 5年間同一担当
- 設計書レビューでは有効な欠陥指摘が少ない
  - 誤字脱字衍字系指摘が73%

文章表現の原則・禁則

文章表現の原則	内容
事実準拠の原則	事実に即して記載する。
断定・推測区分の原則	推測を含める場合は事実と分けてはっきりと記載する。
個性化の原則	決まり文句や流行語、一般論的な表現を避け、実際に存在する個別事項を記載する。
共通理解の原則	訴えたいことがそのまま関係者に理解されるように表現する。
具体化の原則	抽象的な表現を避け、どのような状態や結果なのかを具体的に把握できるように記載する。
一文一義の原則	一つの文に一つの内容を記載する。
簡潔性の原則	余計な修飾語や冗長な説明を削り落として簡潔に記載する。

禁則	悪い例	適切な見直し方法
体言止め・紋切型	モラル 計画	内容を具体化して生々しい出来事を記載する。
不足型・不十分型	レビュー不足 テストが不十分	不足・不十分となっている内容を具体的に示す／不足していることで発生している出来事を明確にする。
対策型	□□基準が存在しない	それがないために発生している困った出来事や状態を記載する。
疑問型	○○スキルに問題あり？	疑問に思った経緯や背景・出来事を具体的に記載する。
断定型・推論型	Aさんはやる気がない 最初からムリな計画なのではないか？	そう考えた経緯や背景・出来事を具体的に記載する。

複数の関係する要素が混在している
⇒一件一葉に分割する

リーダに多くのタスクが集中し、多忙を極めているためレビューが実施できないことがある

分割

分割

分割

リーダに多くのタスクが集中している

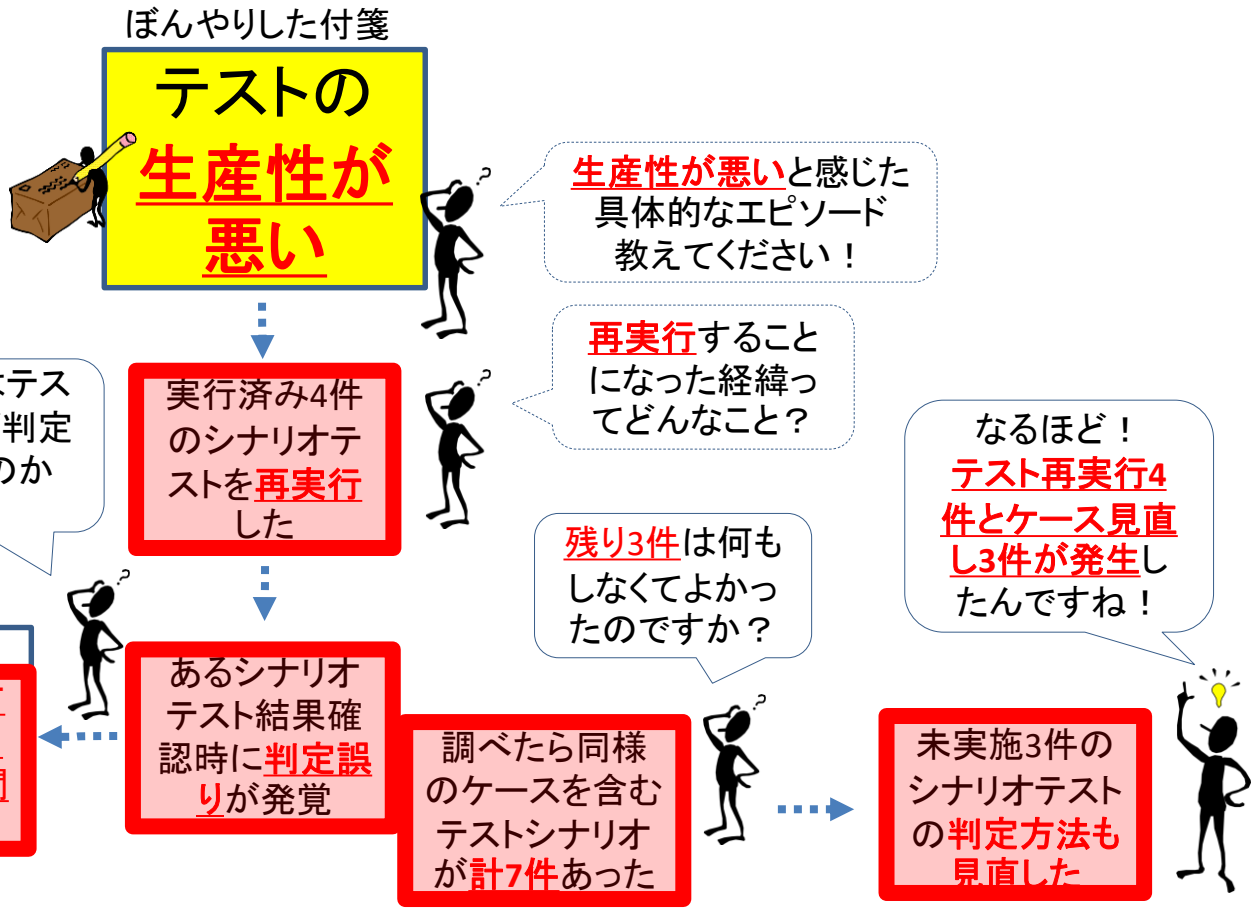
リーダは多忙である

リーダ都合でレビューが実施できないことがある

ぼんやりした(抽象的な)要素は具体化する

共有したい事実情報【結果】

改善検討時に有効な情報
【成果物の状態＝活動結果】



以上の掘り下げにより、 をとりまとめて精査後の要素とします。

不足型(対策裏返し型)要素はそれがないことで何が起きているかを把握する

共有したい事実情報【結果】

改善検討時に有効な情報
【成果物の状態＝活動結果】

不足型記述の付箋

品質基準がない

品質基準がないことで発生している困り事を具体的に教えてください！

リリースしたら障害が多発→対処が大変だった

どんな障害がどのくらい発生したんですか？

なるほど！
リリース後障害7件発生でその対処により、要員が・・・なんだー！

ということは、対処した要員のメインタスクは・・・？

復旧担当者のAさんは2日間泊り込みで対処／他の方も残業爆増

復旧対処した要員のメインタスクがすべてSTOPした

数日機能停止が3件、数時間で復旧したのが4件

それらはテストをPASSしていたのですか？

テスト漏れが4件、テスト済なのに障害になったのが3件あった

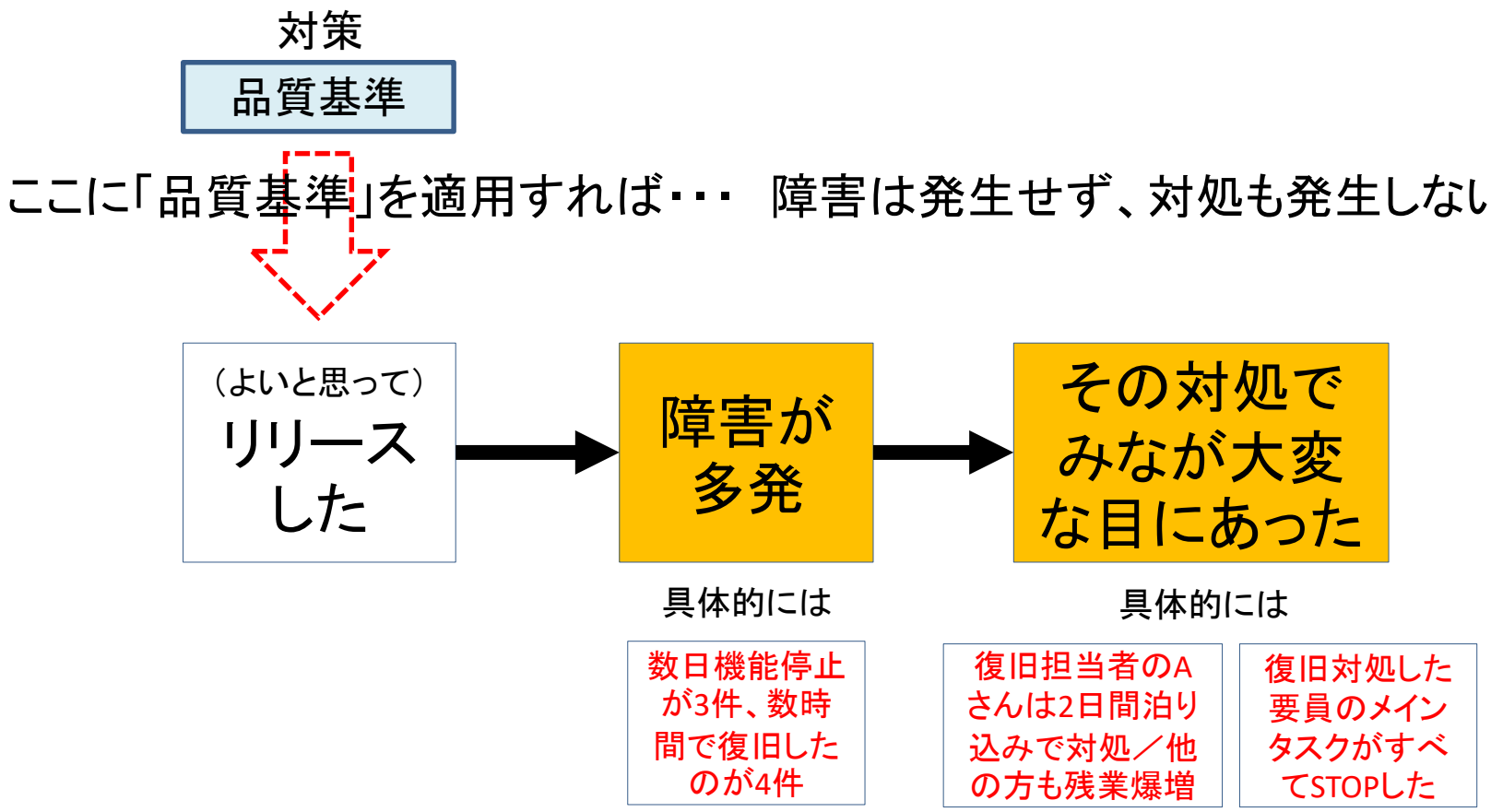
テストケース作成とその内容OKはどうやってたんでしょう？

仕様も断片しかなく、テストケース作成は担当者任せ

※さらに状況確認が必要

以上の掘り下げにより、 をとりまとめて精査後の要素とします。

「～がない」「～が不足している」 の構造



掘り下げの意味と注意事項

- 相手のふりかえり結果の内容を**自分がリアルに理解する**ことを通じて、チームとしてモノゴトの解像度を上げて共有する。

  ※メンバーのふりかえり結果から、自分の、そしてチームのリアルな気づきや共通認識（相互理解）を獲得するための機会を作る

- 詰問や批判、ダメななぜなぜ分析のようにならないでね。❤

  – **主語は「私」**

    例：（私が）ちゃんと理解したいので教えてください！

  – **自分の言葉に言い換えて説明しなおし、確認する**

    例：なるほど！〇〇〇□□□ということですね？　おかげで理解できました！

SaPIDファシリテーター
“プロセスをリードする”

- 有機的リーダシップは、プロセスをリードする。プロセスをリードするとは、人に反応して行動し、彼らに選択を委ね、彼らに自分自身を支配させることである。人々はちょうど庭師がタネに力を与えるのと同じやり方で力を与えられる。すなわち、成長せよと強制する代わりに、彼らのうちに眠っている力をくみ出すのである。

「スーパーエンジニアへの道－技術リーダシップの人間学」
G･M･ワインバーグ著　木村泉訳　より

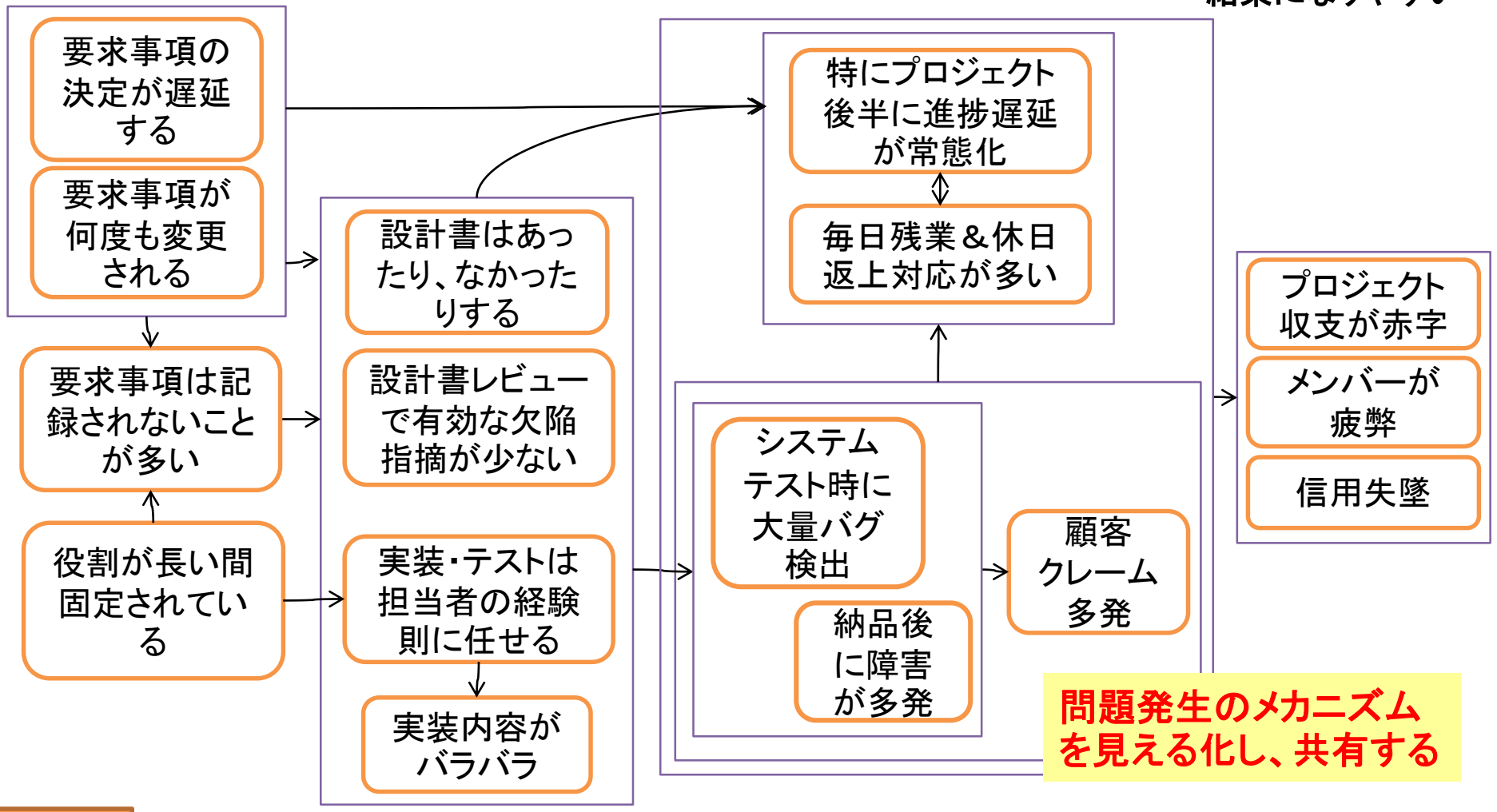
STEP3：問題分析・構造化

【要約】顧客の言いなりで要求事項が不明確なままプロジェクトを進めているため、途中から仕様変更や進捗遅延が多発し、レビューが追い付かず、テストで大量のバグが検出され、納品後クレームが多発している。

摘要

原因 → 結果

原因が存在すると結果になりやすい



P.55改

STEP3-1：各要素を1サイクル上に配置する

①〇〇〇〇〇
サービス

②顧客：
□□□□□
▽▽▽▽

③提供している
価値□□□

④評価方法：
□□□□□
△△△△△

サービス開始

サービスの1サイクルの開始～終了まで(t)

サービス終了

役割が長い間固定されている

特にプロジェクト後半に進捗遅延が常態化

毎日残業＆休日返上対応が多い

納品後に多くの障害が発生

要求事項の決定が遅延する

設計書はあったり、なかったりする

各要素の発生･認知タイミングにやんわり配置する

要求事項が何度も変更される

設計書レビューでは有効な欠陥指摘が少ない

実装･テストは担当者の経験則に任せている

システムテスト時に大量バグ検出

顧客クレーム多発

要求事項は記録されないことが多い

実装内容がバラバラ

プロジェクト収支が赤字

STEP3-2：要素間の因果関係を分析する

要因 → 結果

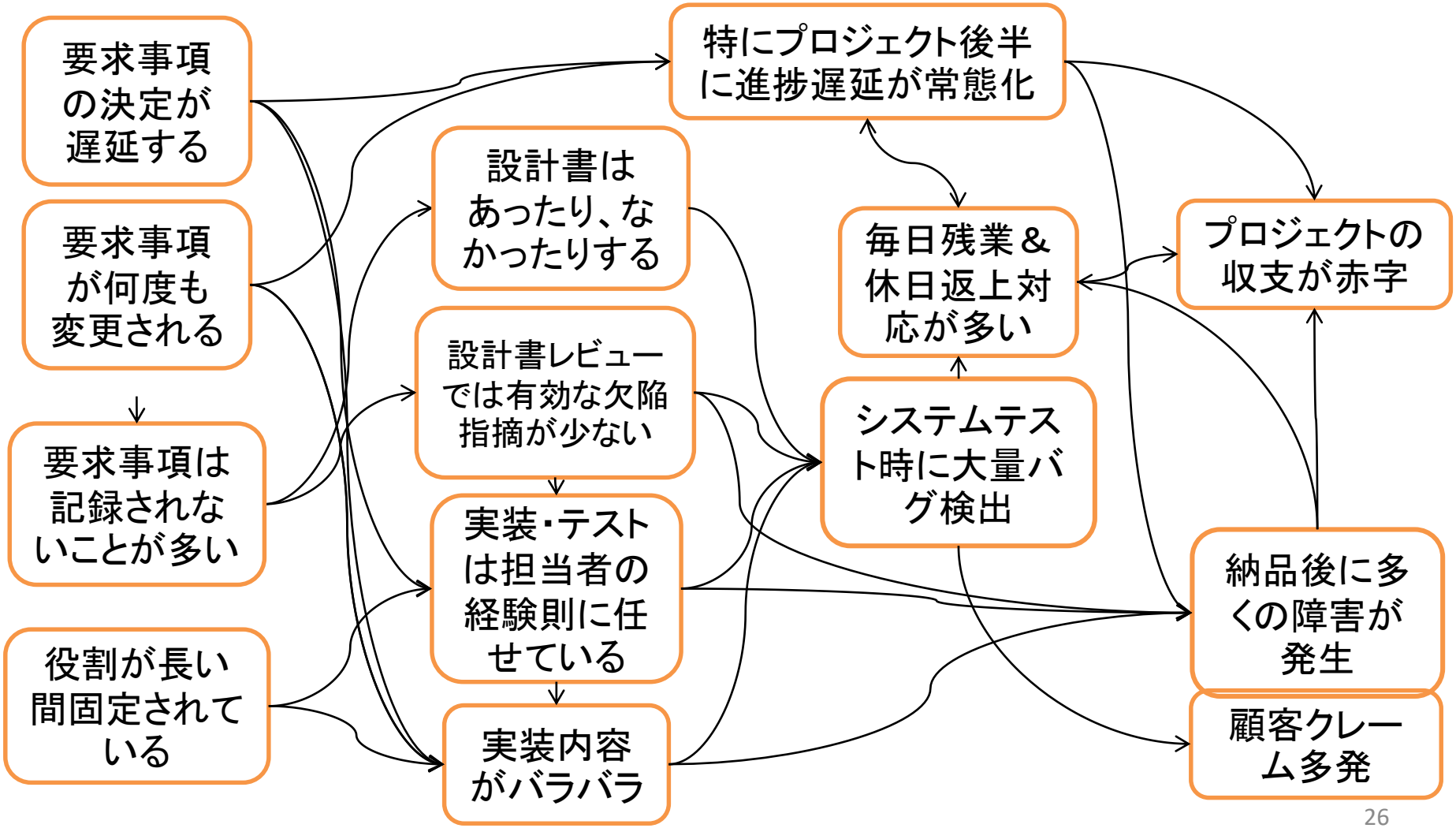
①〇〇〇〇〇
サービス

②顧客：
□□□□□
▽▽▽▽

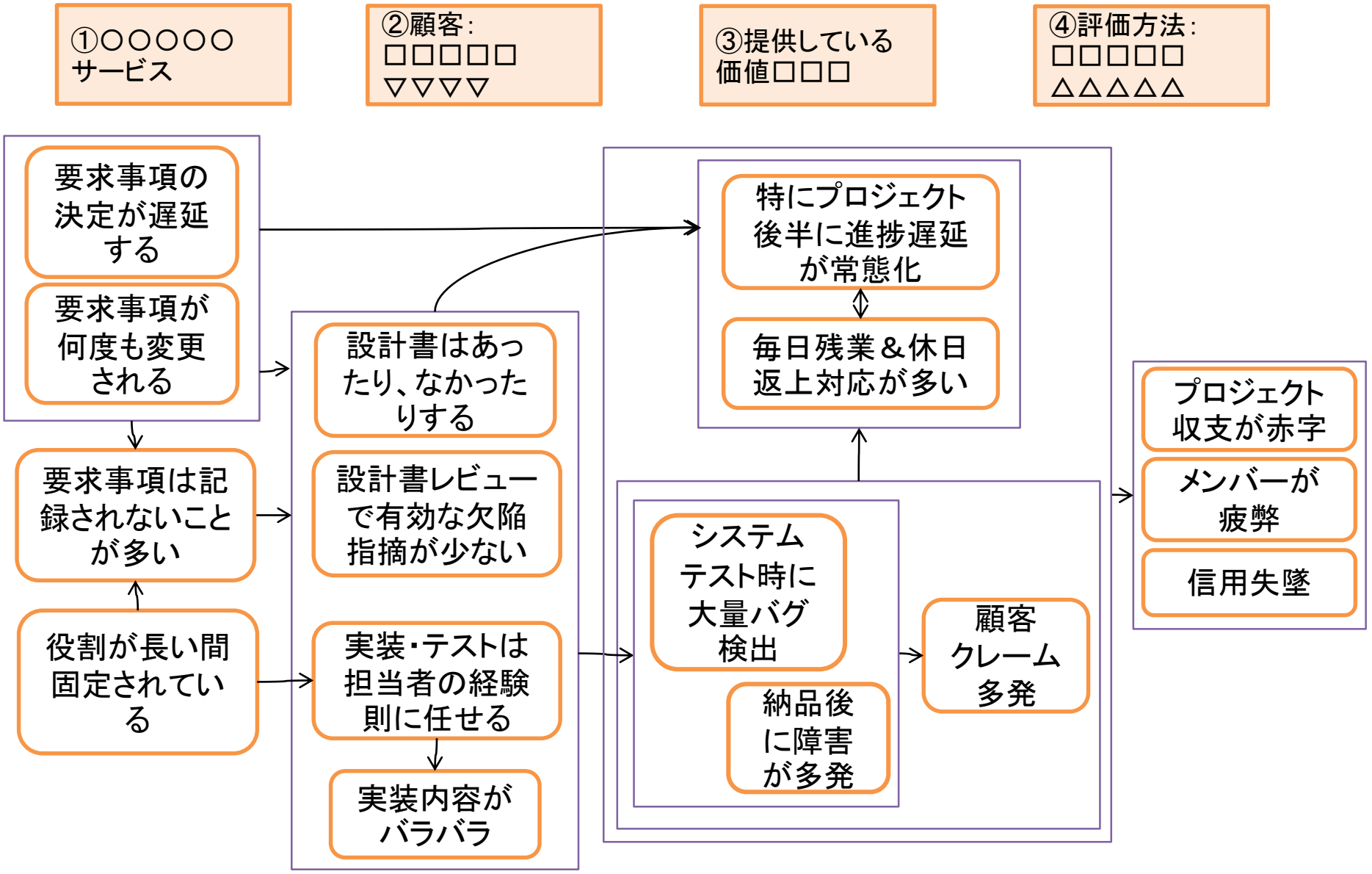
③提供している
価値□□□

④評価方法：
□□□□□
△△△△△

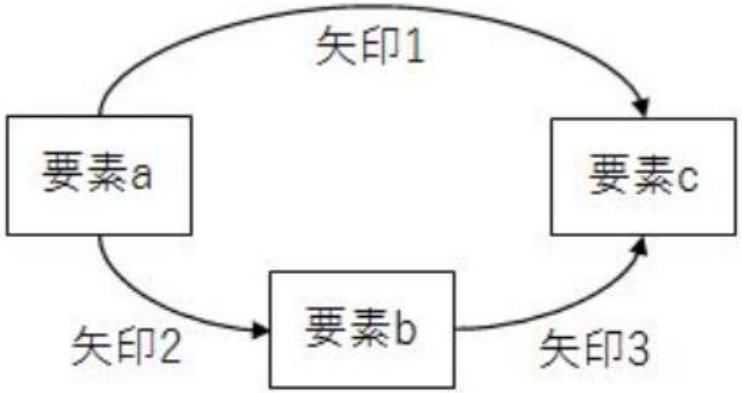
構造化の最中に新しい要素を追加することもある



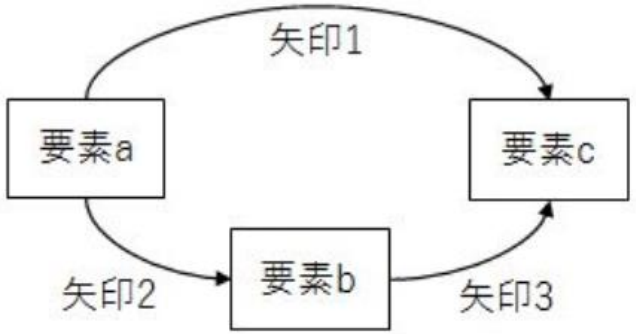
STEP3-3：要素間の関係や内容を簡潔に整理する



関係線(の意味)を正しくつなぐ



ある方が要素間の因果関係を
分析して→を付与した結果

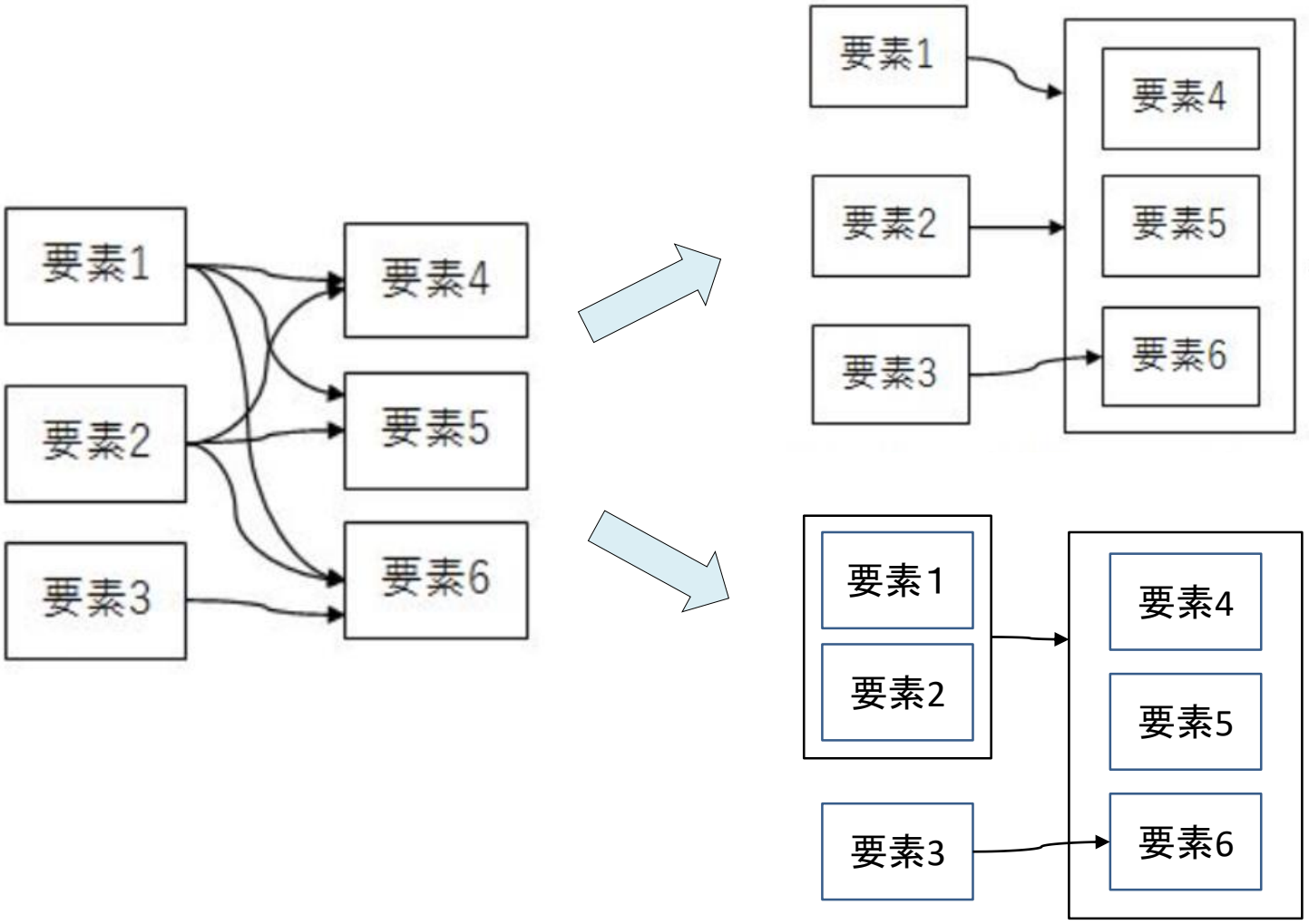


要素aから矢印2・要素b・矢印3を経由して要素cに至る因果関係とは別に、要素aから直接要素cにつながる関係性は存在するとわかった場合
(構造図はそのまま)

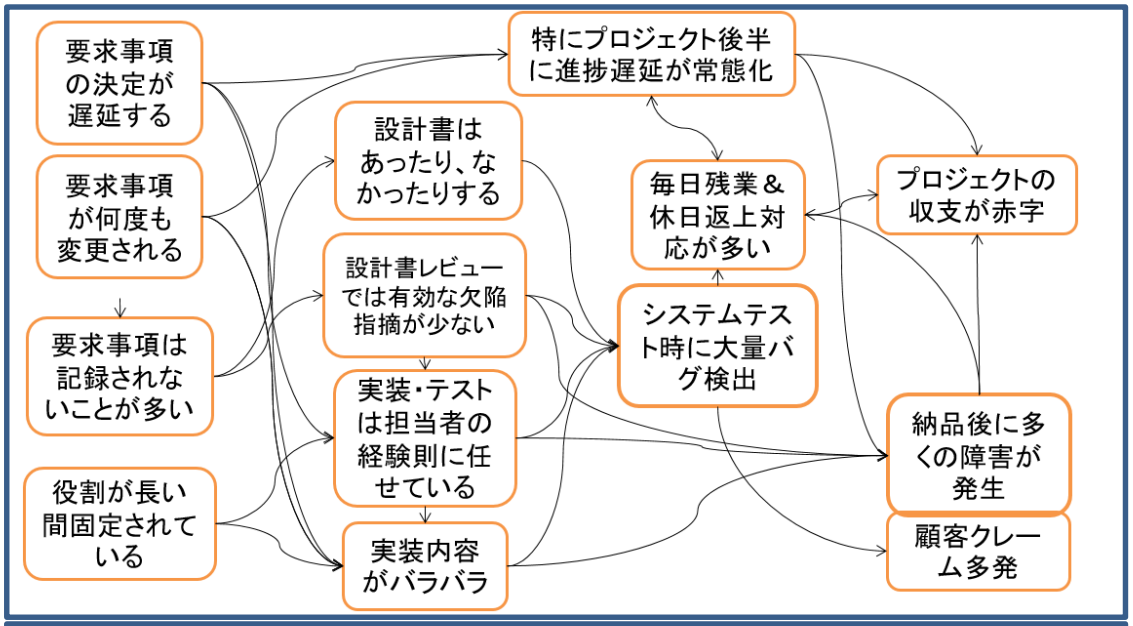
要素aから矢印2・要素b・矢印3を経由して要素cに至る因果関係のみが正しいという結果になった場合
(矢印1を削除する)

関係線は可能な限り少なく/クロスしないように

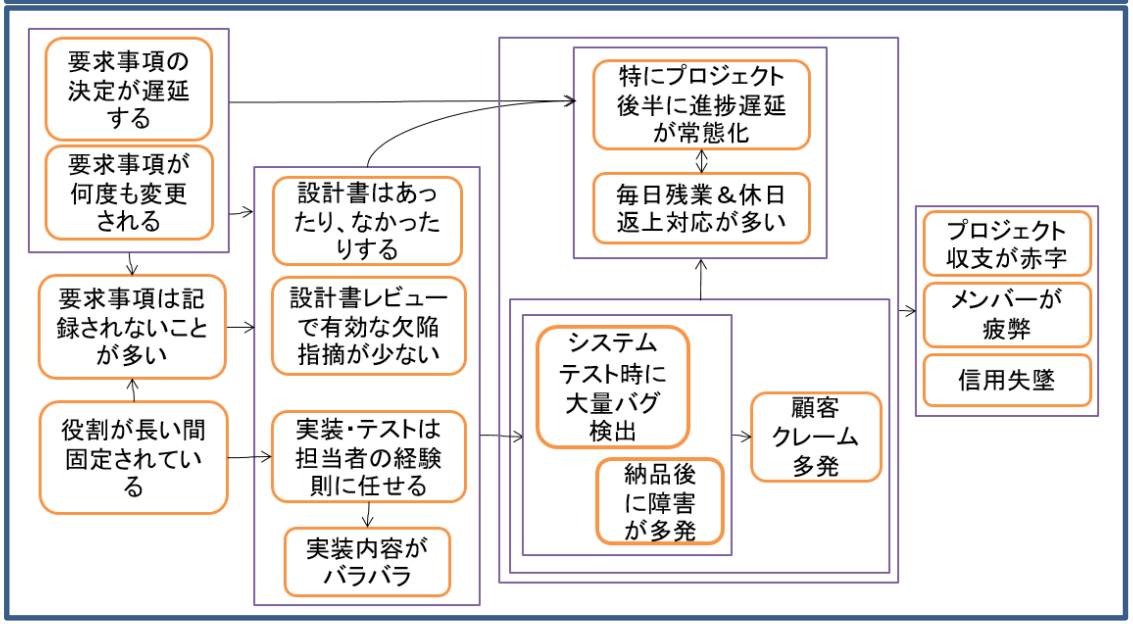
関係線が最小･クロスしない＝わかりやすい＝解きやすい



どちらがわかりやすいか？

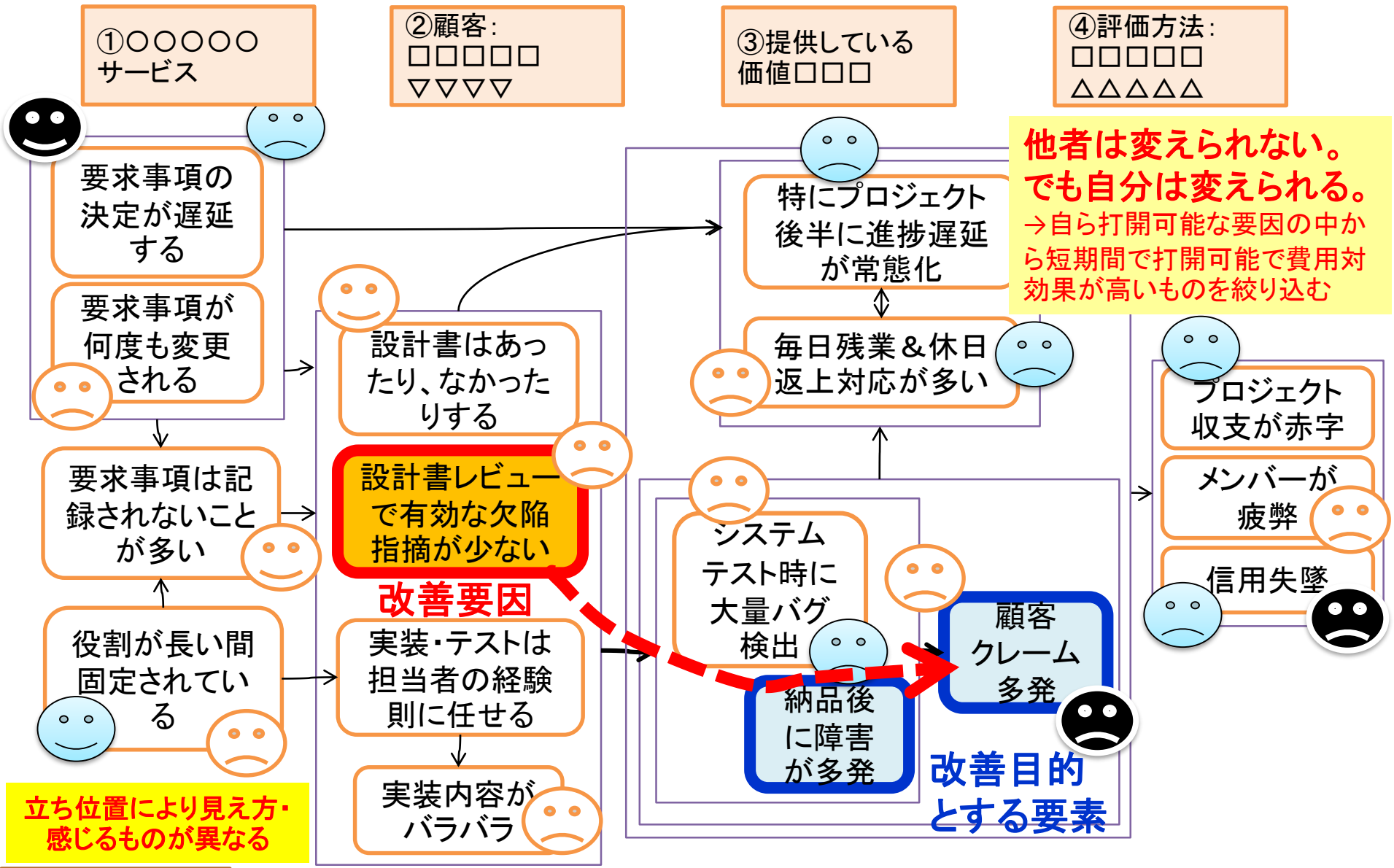


- 関係線が多い
- クロスする線が多い
- 発生時の重なりが多い



- 関係線が少ない
- クロスする線がない
- 発生時の重なりが少ない

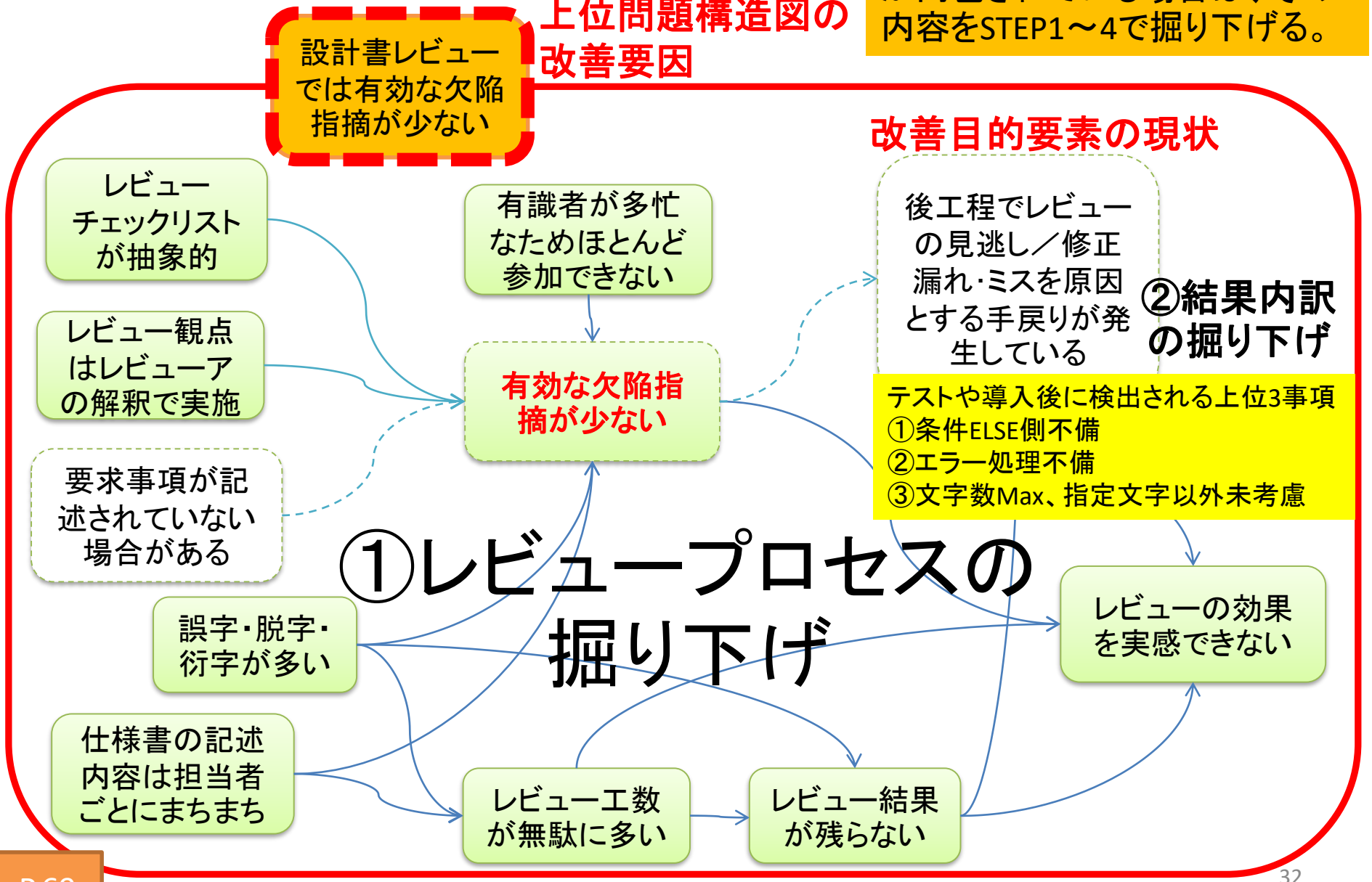
STEP4：改善ターゲット検討・特定



P.61～62改

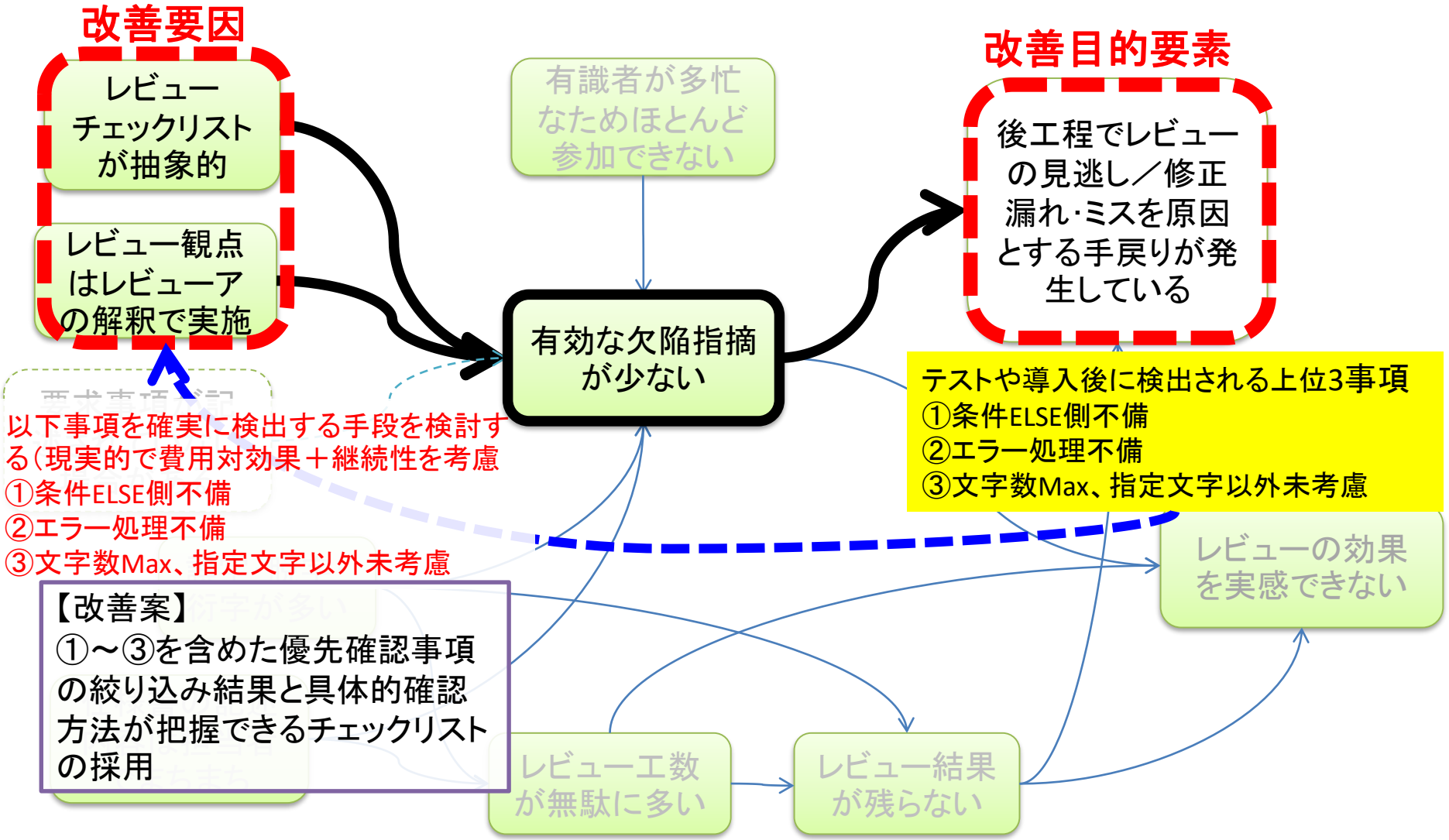
STEP5-1：改善対象の掘り下げ

特定した対象の中に様々な要因が内包されている場合は、その内容をSTEP1～4で掘り下げる。



STEP5-2：改善手段検討・決定

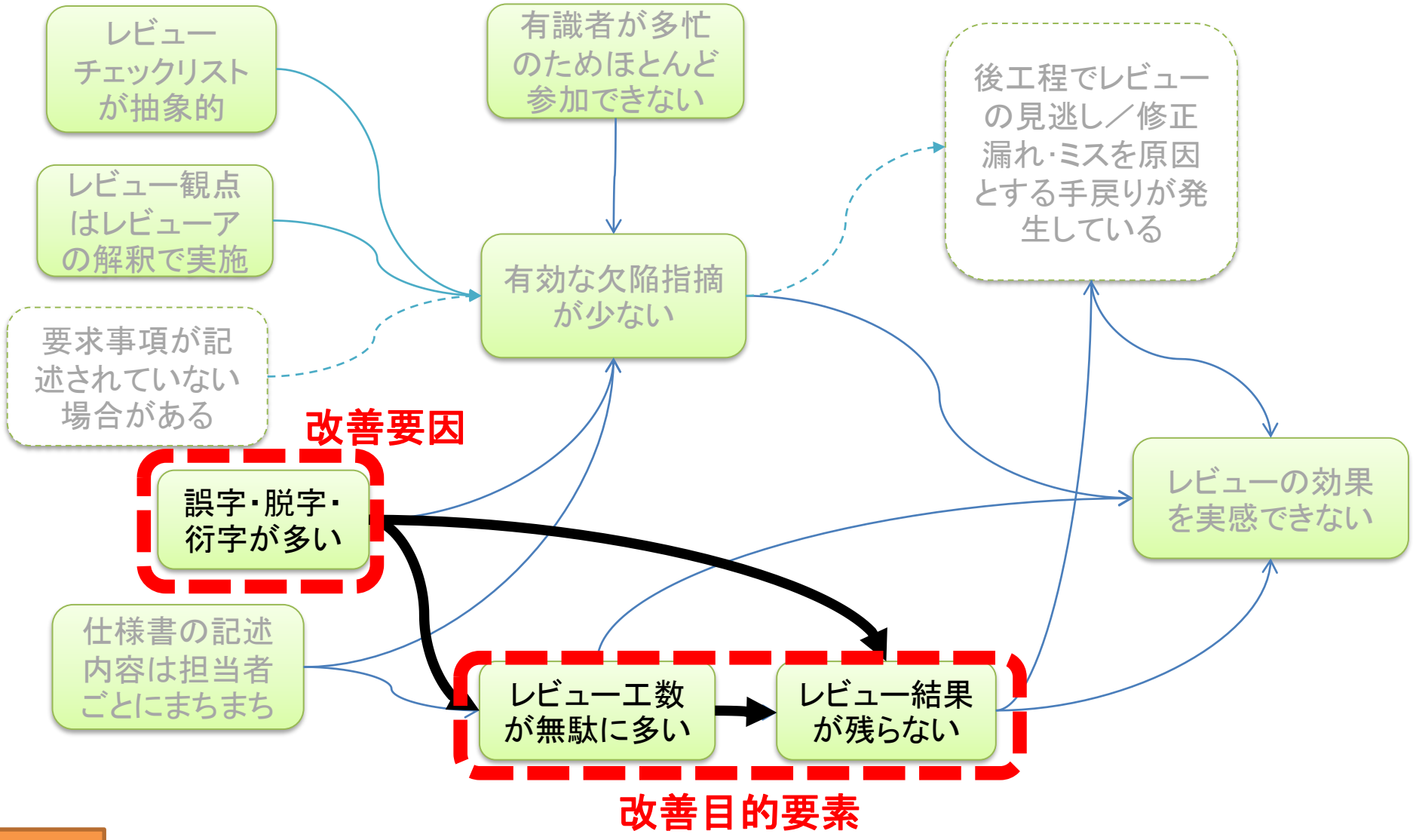
レビューの本質的な改善に着手する例



P.70

STEP5-2：改善手段検討・決定（別解）

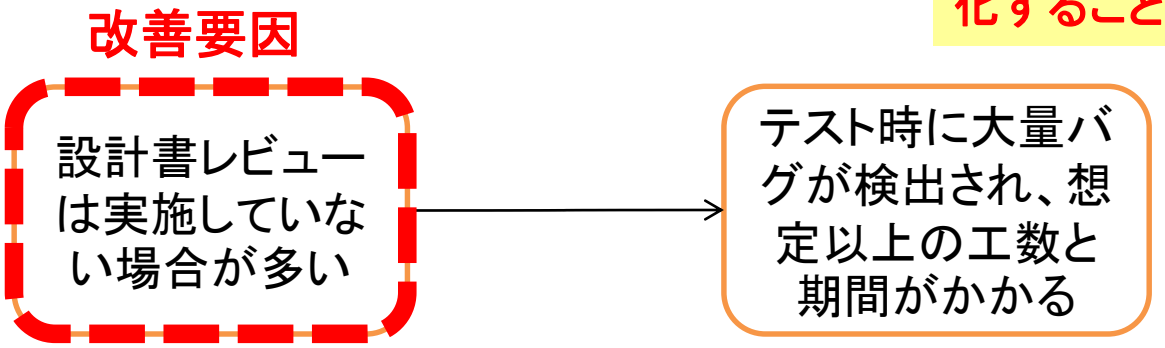
先行して無駄な工数を削減し、余裕を確保してから本質的な改善に着手する例



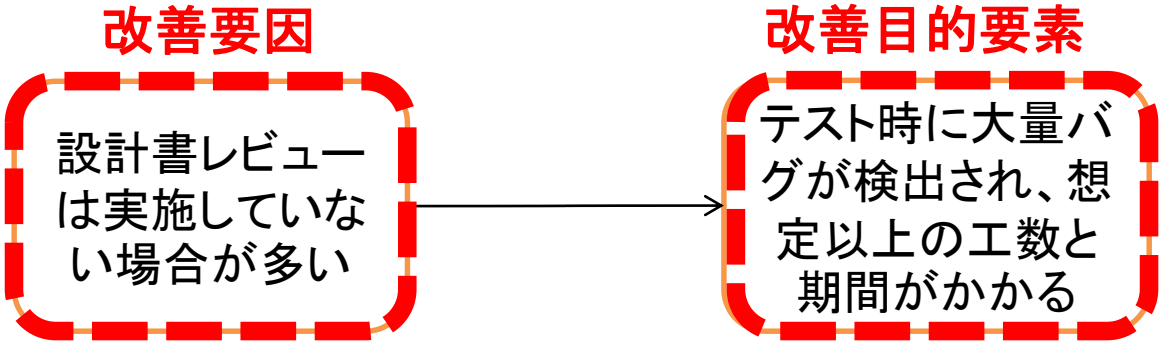
P.72

STEP6：改善目標の検討・決定

改善目標も個人・チーム・組織の状況に応じて段階的に高度化することが重要



施策系改善目標
- 例1：レビュー実施
- 例2：レビュー実施率



施策系改善目標
- 例1：レビュー実施
- 例2：レビュー実施率

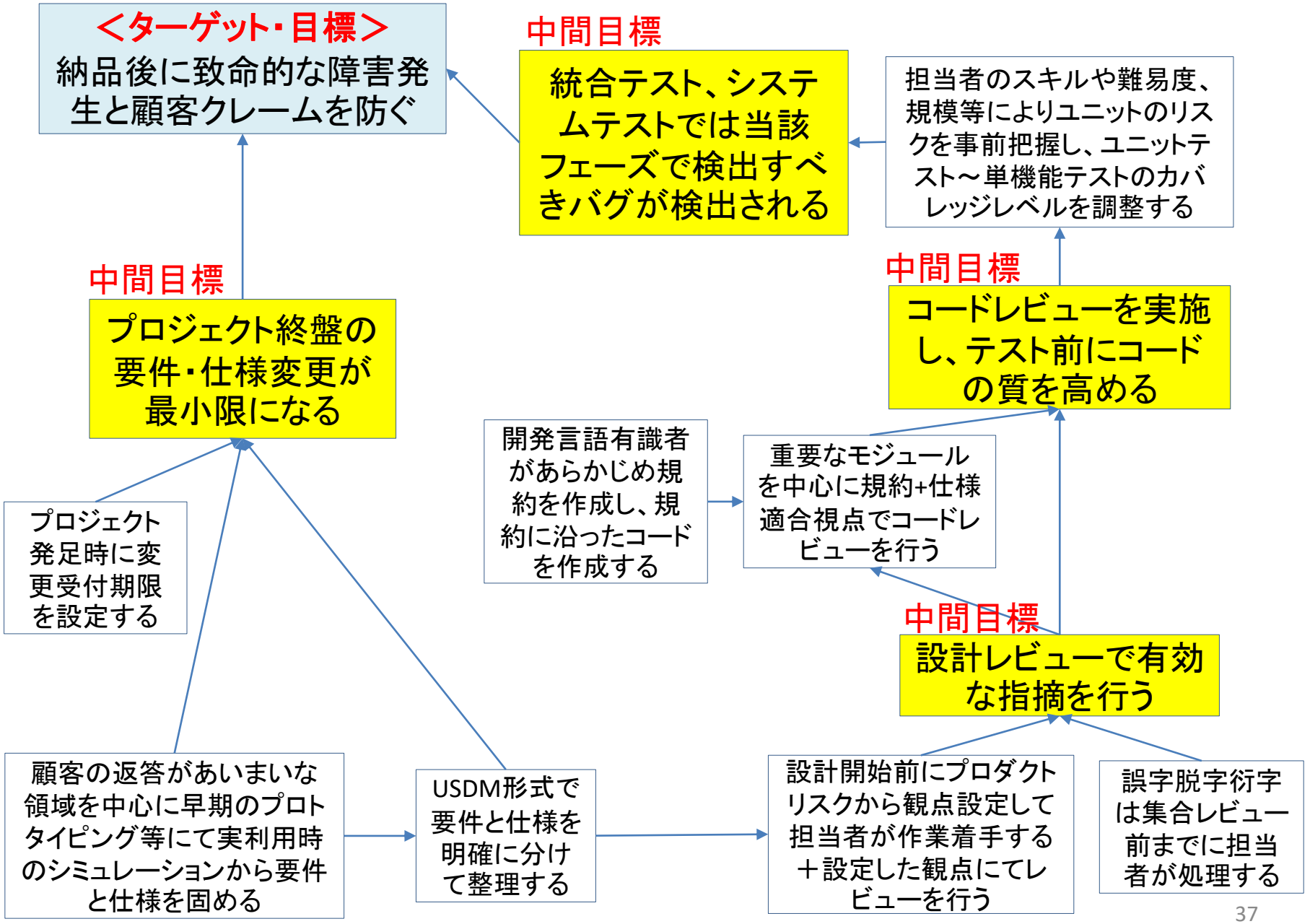
成果系改善目標
- 例1：規模あたりのテスト時バグ検出量減少
- 例2：規模あたりのテスト工数減少
- 例3：規模あたりのテスト期間の短縮

P.78～79

アンビシャスターゲットツリー　1/2

①ターゲット：納品後に致命的な障害発生と顧客クレームを防ぐ		
②障害	③中間目標	④行動
統合テスト、システムテストで大量にバグが検出される	統合テスト、システムテストでは当該フェーズで検出すべきバグが検出される	・担当者のスキルや難易度、規模等によりユニットのリスクを事前把握し、ユニットテスト～単機能テストのカバレッジレベルを調整する
コードレビューが実施されない	コードレビューを実施し、テスト前にコードの質を高める	・開発言語有識者があらかじめ規約を作成し、規約に沿ったコードを作成する ・重要なモジュールを中心に規約+仕様適合視点でコードレビューを行う
設計レビューで有効な指摘ができていない	設計レビューで有効な指摘を行う	・誤字脱字衍字は集合レビュー前までに担当者が処理する ・設計開始前にプロダクトリスクから観点設定して担当者が作業着手する＋設定した観点にてレビューを行う
プロジェクト終盤になって要件や仕様がゴロゴロ変わる	プロジェクト終盤の要件・仕様変更が最小限になる	・USDM形式で要件と仕様を明確に分けて整理する ・プロジェクト発足時に変更受付期限を設定する ・顧客の返答があいまいな領域を中心に早期のプロトタイピング等にて実利用時のシミュレーションから要件と仕様を固める

アンビシャスターゲットツリー　2/2

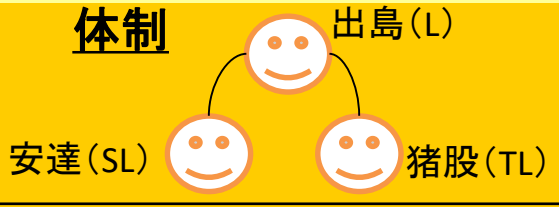


STEP7：改善計画立案

改善計画立案段階で初めて様式に書き始めるのは形式対応になりやすいので注意

チーム名：やってみなはれGr

体制



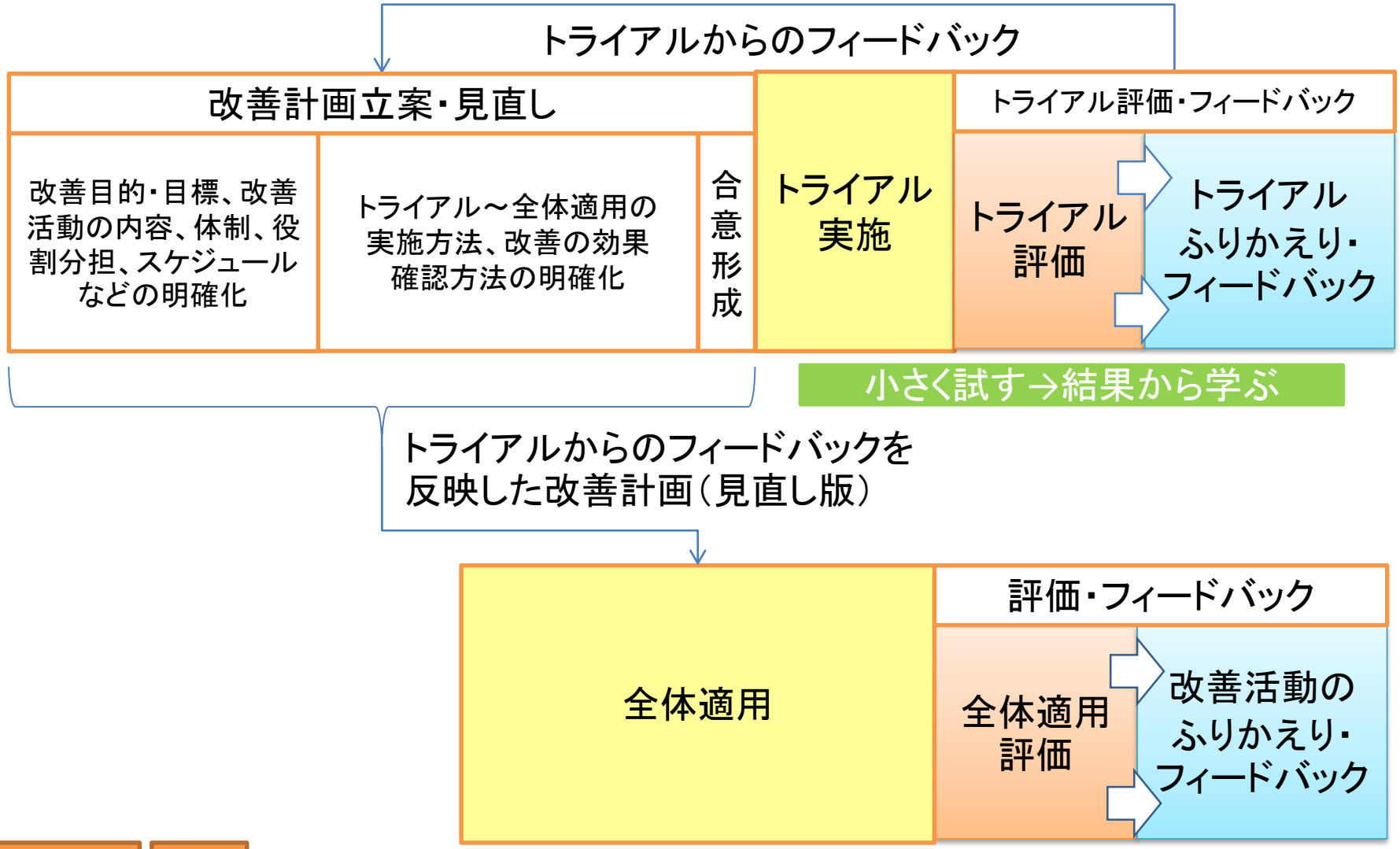
テーマ：〇●運営の効果向上＆効率化を目指す
□■情報作成～共有方法の改善

1.業務概要	△▲部の統括運営・管理を行っています。 部署運営の効果・効率に直結する運営の最適化を目指しています。			
2.着目した問題	・内部、外部関係者間で共有、活用する□■情報の作成・修正・共有に手間と時間がかかっている。共有情報が有効活用されていない。部署内で共有情報活用が進まない理由に情報提供のタイムリーさの欠如が存在している。			
3.要因と改善方法 ※参照：5.改善前後の運営イメージ	要因1：共有情報の作成を元ネタ情報FIX後に手作業で行い、その後再確認、共有、連絡発信の流れになっている。 →改善方法1：元ネタ情報がFIXするまでの過程で同時並行で作成・関係者確認を行う。 要因2：共有情報が紙媒体のため、ロケーションが離れるとアクセスできない。 →改善方法2：情報を格納する共有フォルダを構築し、そこにテキストコピー可能のPDFで格納し、各関係者がアクセスすることにする。			
4.改善期待効果と改善目標	評価指標	現状 直近2カ月間、対象9回の平均	改善目標	実績 （計画時空欄）
	①情報作成～共有・関係者連絡までの期間（日数）	5～6日間	0日間（当日完了）	0日間 1.5時間程度
	②手戻り工数 （人時）	・問合せ14件 ・対応工数7.5人時	0件、0人時に近づける	2件　3分程度
	③共有情報活用率	39%	80%以上	82%

様式記載例

P.84

STEP8：改善トライアルと評価・フィードバック
STEP9：全体適用と評価・フィードバック



P.87～94
P.92

STEP9：全体適用→評価・ふりかえり＆フィードバック

全体適用

評価・ふりかえり
によるフィードバック

フィードバック事項
の反映→改善継続