

サピッド SaPIDの概要

株式会社 HBA Software Quasol 安達 賢二

adachi@hba.co.jp

<http://www.softwarequasol.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。

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

SaPIDが解決したい世界

～システム思考×デザイン思考実践により依存から能動・自律へ

プロセスモデルベース トップダウン偏重
やらせる／やらされるプロセス改善の問題

CMMI

ISO9001



問題構造図
(因果関係モデル)



システム
思考



デザイン
思考

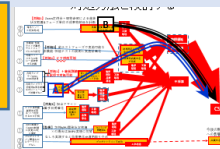
未来予想図



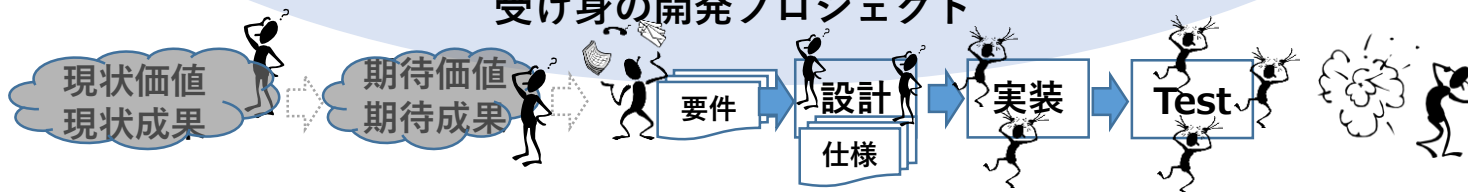
管理統制ばかりでぶら下がりメンバーによる
問題駆動/強制・指示型プロジェクト運営



FRIモデル



突然要件や仕様から始まる
受け身の開発プロジェクト



SaPID Process Model

STAGE 0 ビジネス価値向上～新ビジネス創出実践

STEP 1 ビジネス価値共有

STEP 2 ビジネス価値向上

STEP 3 新ビジネス創出

STAGE 1
現状
把握

STEP 1
問題洗い出し・
引き出し

STEP 2
事実確認・
要素精査

STEP 3
問題分析・
構造化

STAGE 2
改善の
検討

STEP 4
改善ターゲット
の検討・特定

STEP 5
改善策の検討・
決定

STEP 6
改善目標の
検討・決定

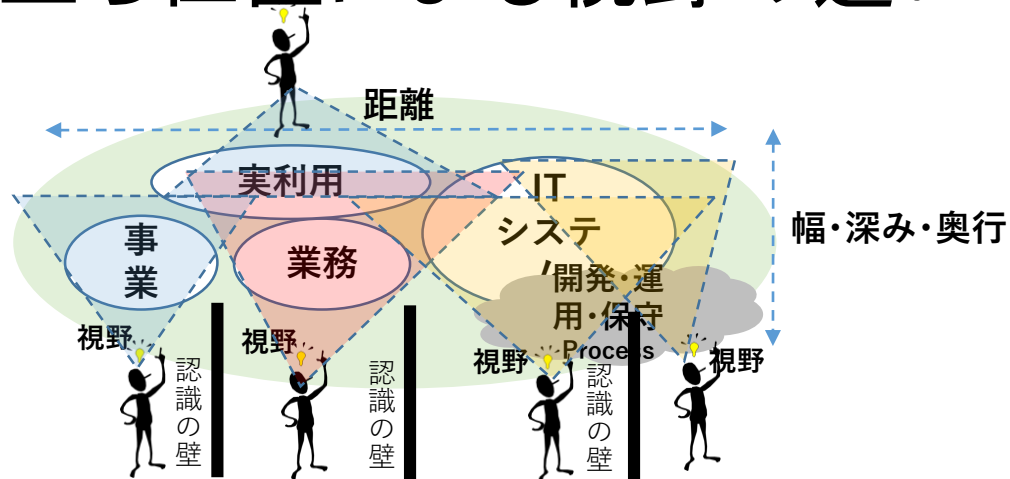
STAGE 3
改善の
実行

STEP 7
改善計画
立案

STEP 8
改善トライアルと
評価・フィードバック

STEP 9
全体適用と
評価・フィードバック

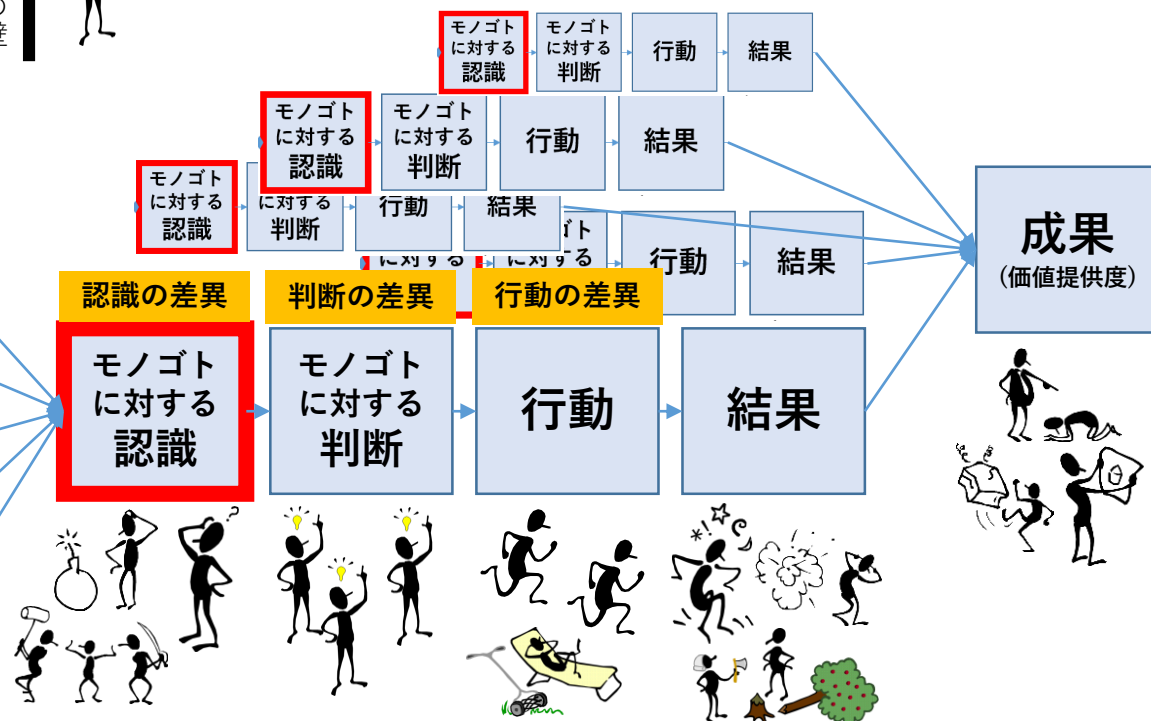
立ち位置による視野の違い等による認識不一致



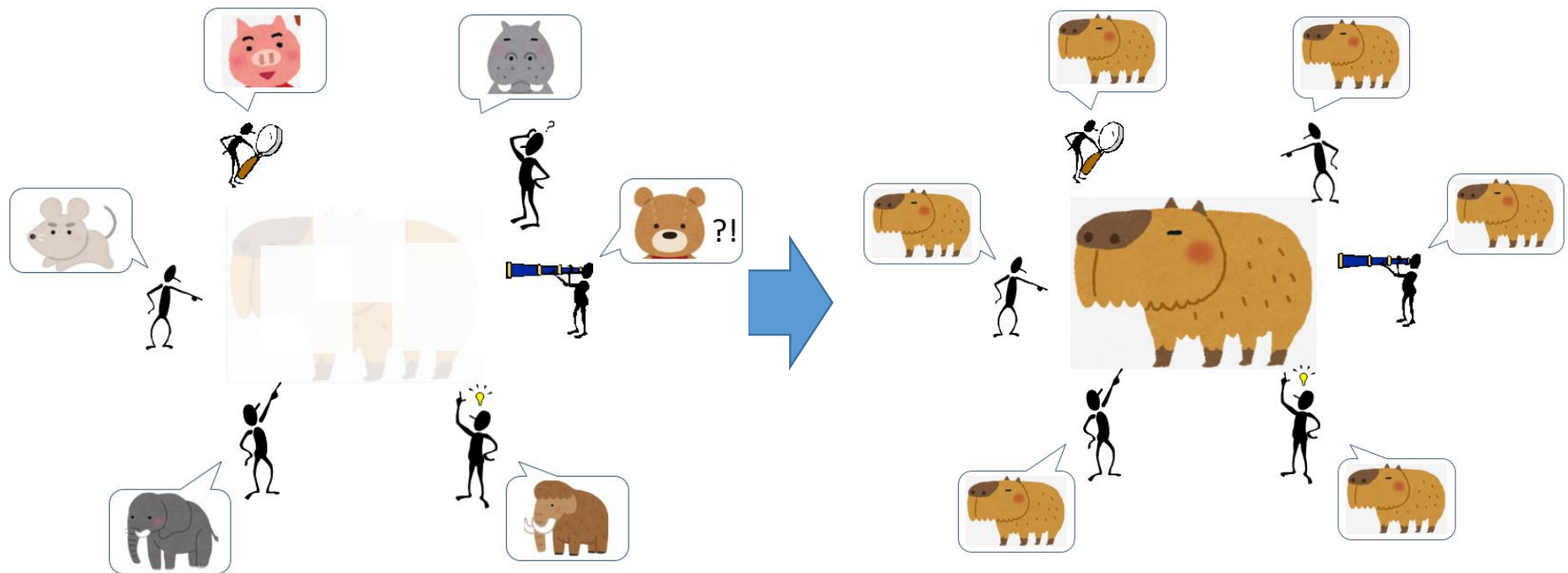
認識の壁によるサイロ化

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？

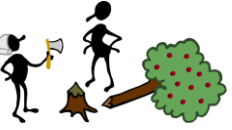
目的を達成するために、1つまたは複数の構成要素が相互に連携・作用するしくみ全体のこと

- 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！ 

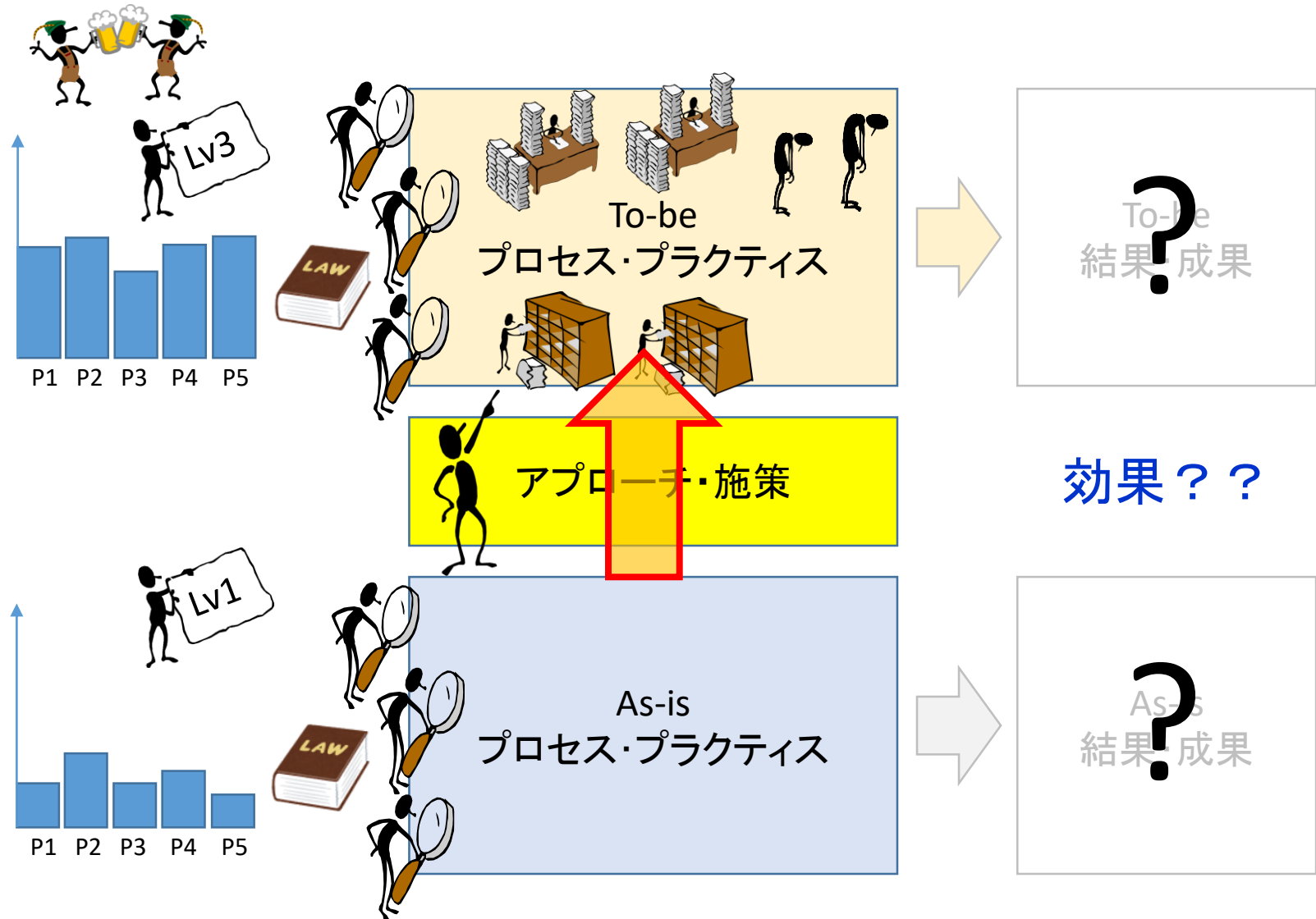
引用元: -the d.school bootcamp bootleg

【4つのMindset】

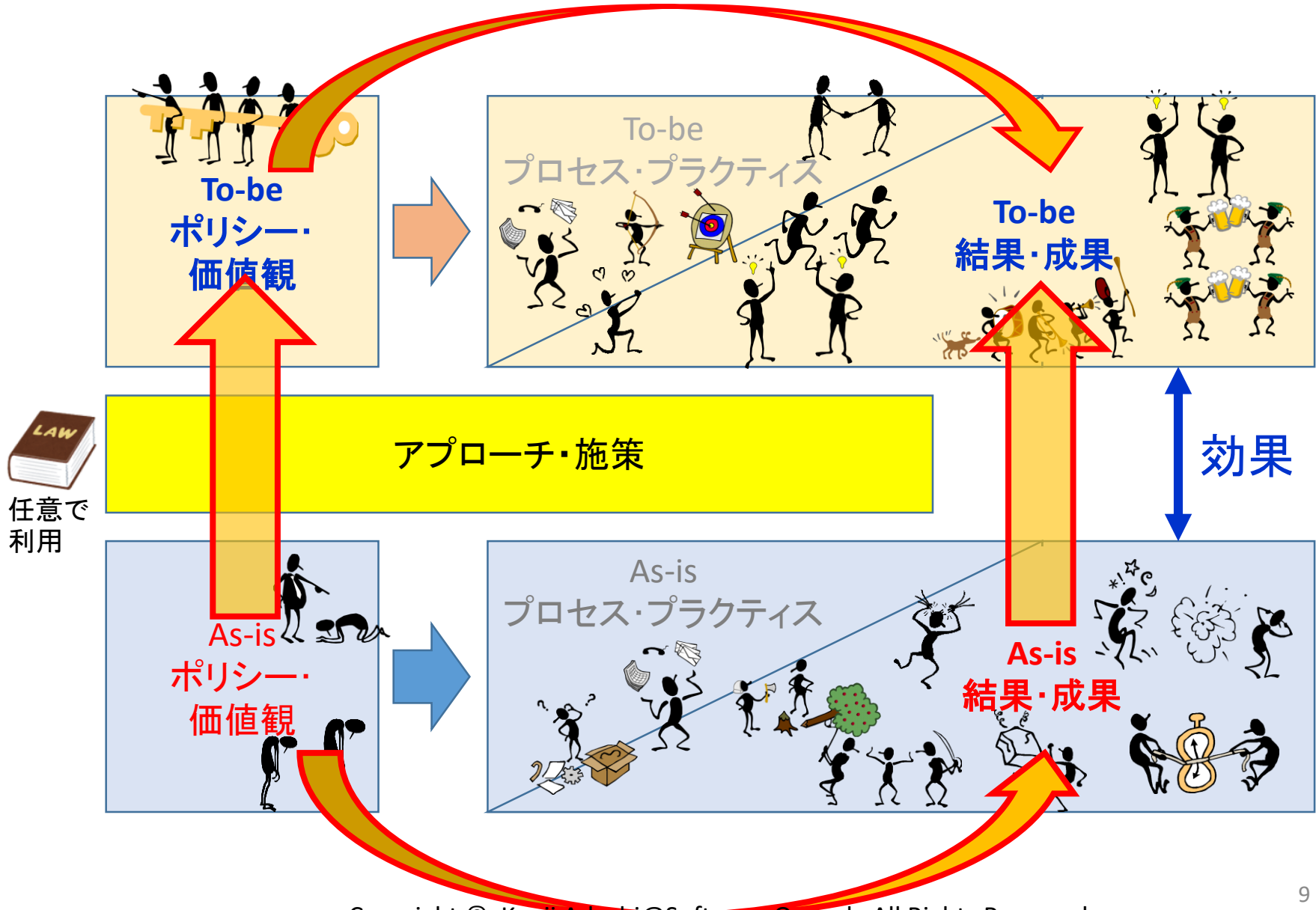
- Human-Centered: **人間中心**
人間を意識して考える 
- Collaborative: **協力的**
多様性を活かす 
- Optimistic: **楽観的**
われわれにもできる！信念 
- Experimental: **実験的** 
たくさんの試行経験から学ぶ

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りわり決めます。
 - 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
- 今回のテーマは、以下とします。
- 「自分の仕事・役割の実行と成果における問題点」**

STAGE0(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

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?

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

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ります。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○○○○○
○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

STAGE0(STEP0) ふりかえり結果

Keep	Try
Problem	

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

顧客クレーム多発

プロジェクトの収支が赤字

要求事項の決定が遅延する

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

協力会社Aは危ないのではないかと

システムテスト時に大量バグ検出

開発標準や各種判定基準がない

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

設計書レビューでは有効な欠陥指摘が少ない

実装内容がバラバラ

設計書はあったり、なかったりする

統合テスト・システムテストで想定していた以上の(はるかに超える)バグが検出された

設計レビューでは有効な指摘が少ない(誤字・脱字・衍字が多い)

顧客からシステムへの要求事項がなかなか提示されなかった

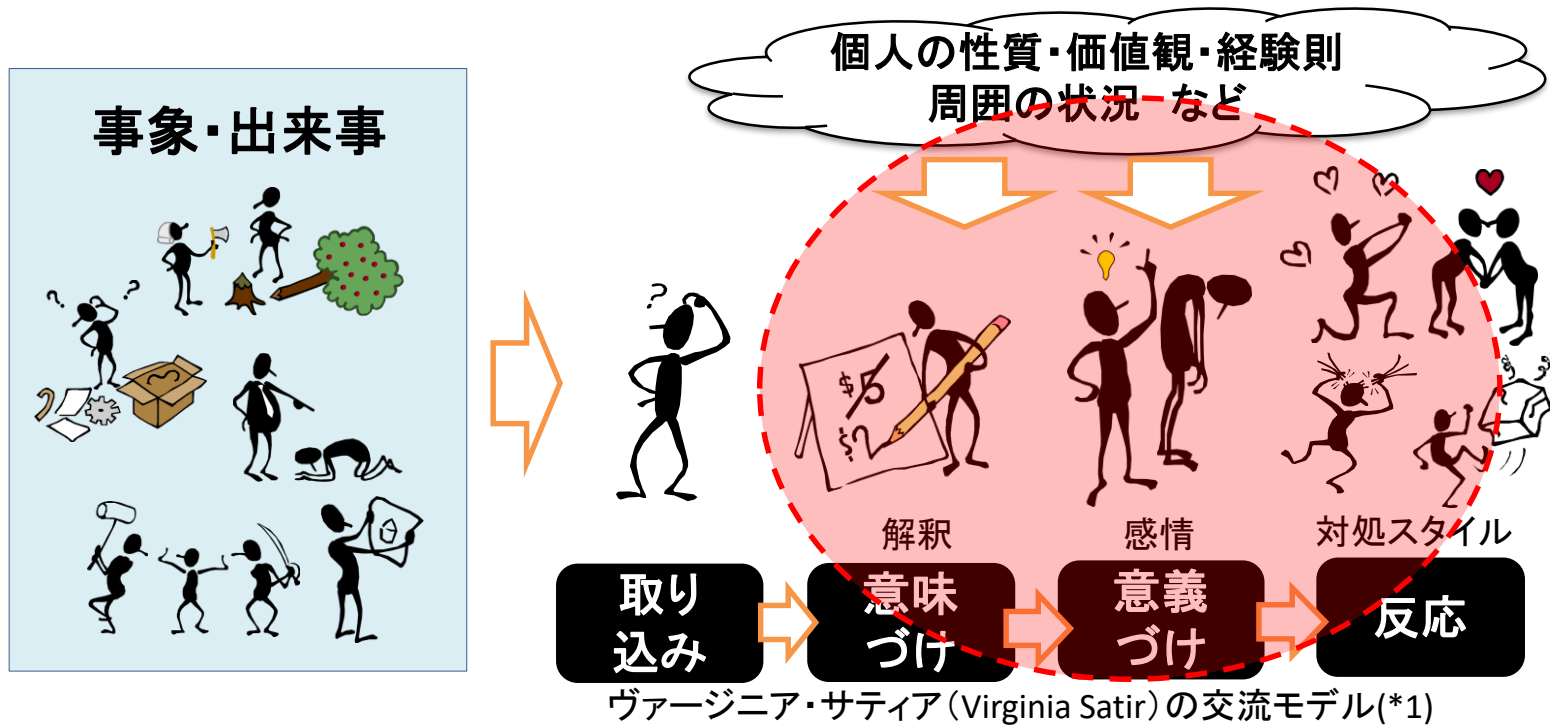
良い・悪いは抜きにして、何が起きているのか、どう感じているのか等をありのまま収集する

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



こんなことが
あった！

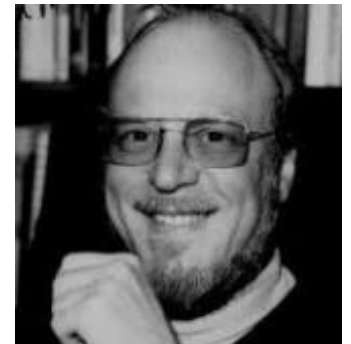
うれしい！
イヤだなあ...

*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。

STEP1 ふりかえり結果

Keep	Try
Problem	

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	それがなかった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	〇〇スキルに問題あり？	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

リーダに多くのタスクが集中している

分割

リーダは多忙である

分割

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】

ぼんやりした付箋

テストの
生産性が悪い

生産性が悪いと感じた
具体的なエピソード
教えてください！

再実行すること
になった経緯っ
てどんなこと？

残り3件は何も
しなくてよかつ
たのですか？

なるほど！
テスト再実行4
件とケース見直
し3件が発生し
たんですね！

ケース判定方法
記述が誤ったの
は仕様記述の問
題ですかね？

判定誤りはテス
ト担当者が判定
を誤ったのか
な？

実行済み4件
のシナリオテ
ストを再実行
した

テスト担当者は
ケースの判定方
法に従ったが判
定方法記述が間
違っていた

あるシナリオ
テスト結果確
認時に判定誤
りが発覚

調べたら同様
のケースを含む
テストシナリオ
が計7件あった

未実施3件の
シナリオテスト
の判定方法も
見直した

仕様記述が
複数の解釈
が成り立つ
状態

※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。

不足型(対策裏返し型)要素はそれがない ことで何が起こっ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！

どんな障害がど
のくらい発生し
たんですか？

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？

それらはテスト
をPASSしていた
のですか？

リリースしたら
障害が多発→
対処が大変
だった

ということは、対
処した要員のメイ
ンタスクは…？

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！

仕様も断片し
がなく、テスト
ケース作成は
担当者任せ

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

数日機能停止
が3件、数時
間で復旧した
のが4件

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

※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



品質基準がない
のが問題！

※SaPIDでは、このような付箋(情報)を「問題」ではなく「解決手段の提案」として認識し、**実際に存在する事象や状態**を把握する

対策

品質基準

ここに「品質基準」を適用すれば・・・ 障害は発生せず、対処も発生しないはず



(よいと思って)
リリース
した

障害が
多発

その対処で
みなが大変
な目にあった

具体的には

数日機能停止
が3件、数時
間で復旧した
のが4件

具体的には

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

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！○○○□□□ということですね？ おかげで理解できました！

事実^に立脚する～何も足さない何も引かない
相手の好き嫌いや自分の価値観／色眼鏡で判断しない

SaPIDファシリテーター

“プロセスをリードする”

- 有機的リーダーシップは、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－技術リーダーシップの人間学」

G・M・ワインバーグ著 木村泉訳 より

STEP2 ふりかえり結果

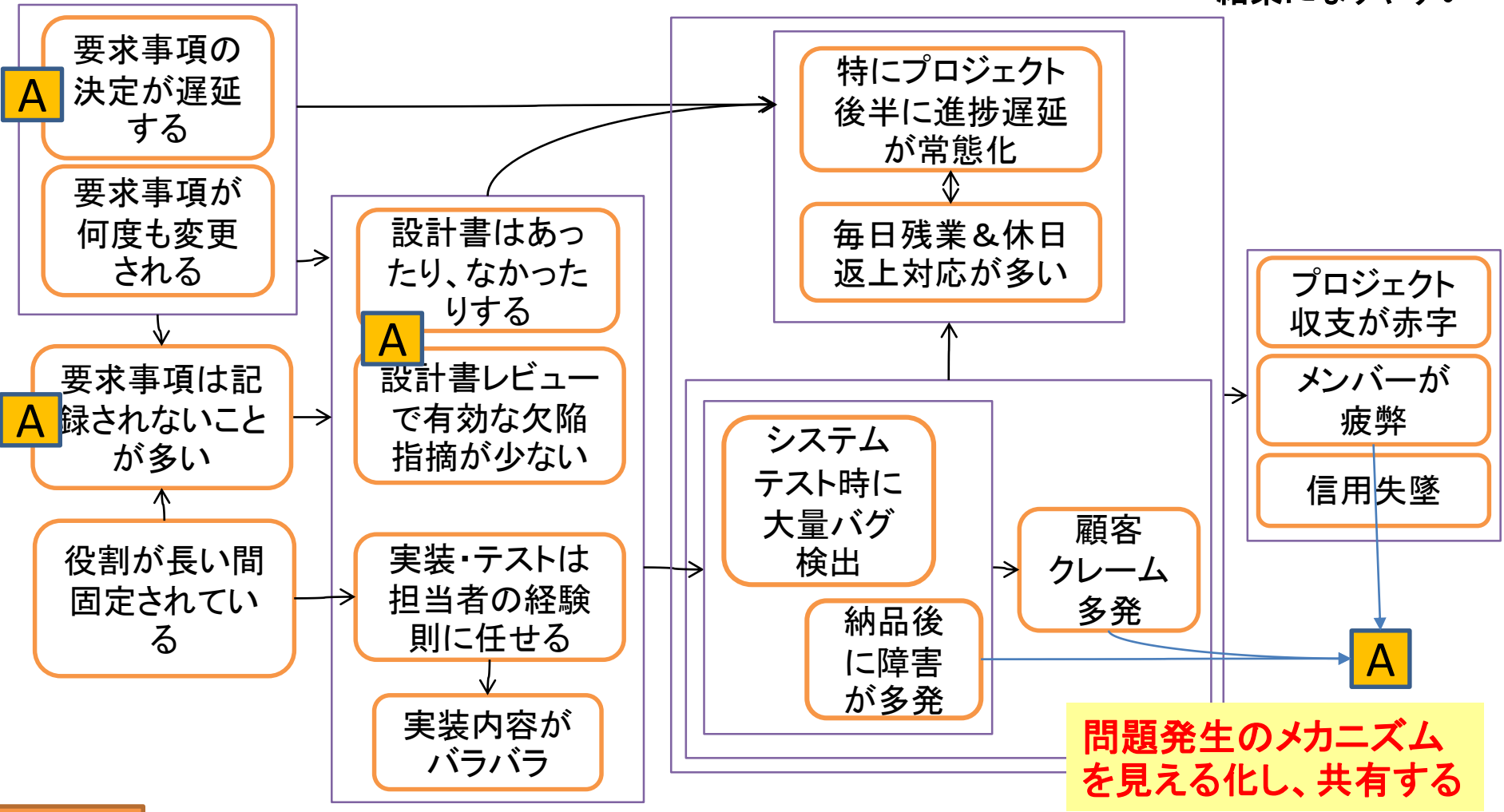
Keep	Try
Problem	

STEP3: 問題分析・構造化

摘要

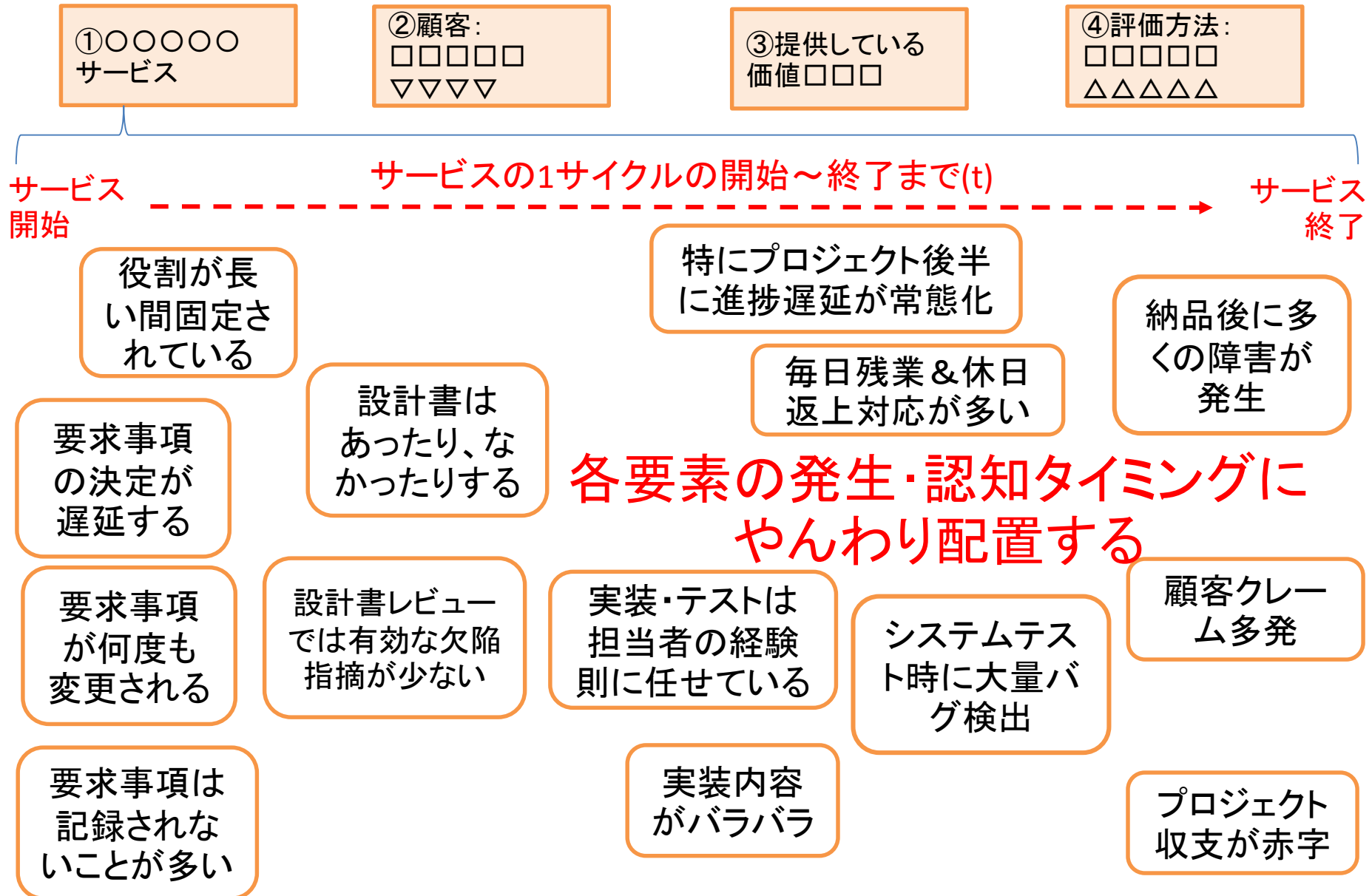
【要約】顧客の言いなりで要求事項が不明確なままプロジェクトを進めているため、途中から仕様変更や進捗遅延が多発し、レビューが追い付かず、テストで大量のバグが検出され、納品後クレームが多発している。

原因 → 結果
原因が存在すると結果になりやすい



問題発生メカニズム
見える化し、共有する

STEP3-1: 各要素を1サイクル上に配置する



時系列配置ができる要素と 配置しにくい要素

要素配置の工夫例

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

管理的な要素を配置

例：見積・計画→進捗管理→終了判定→ふりかえりなど

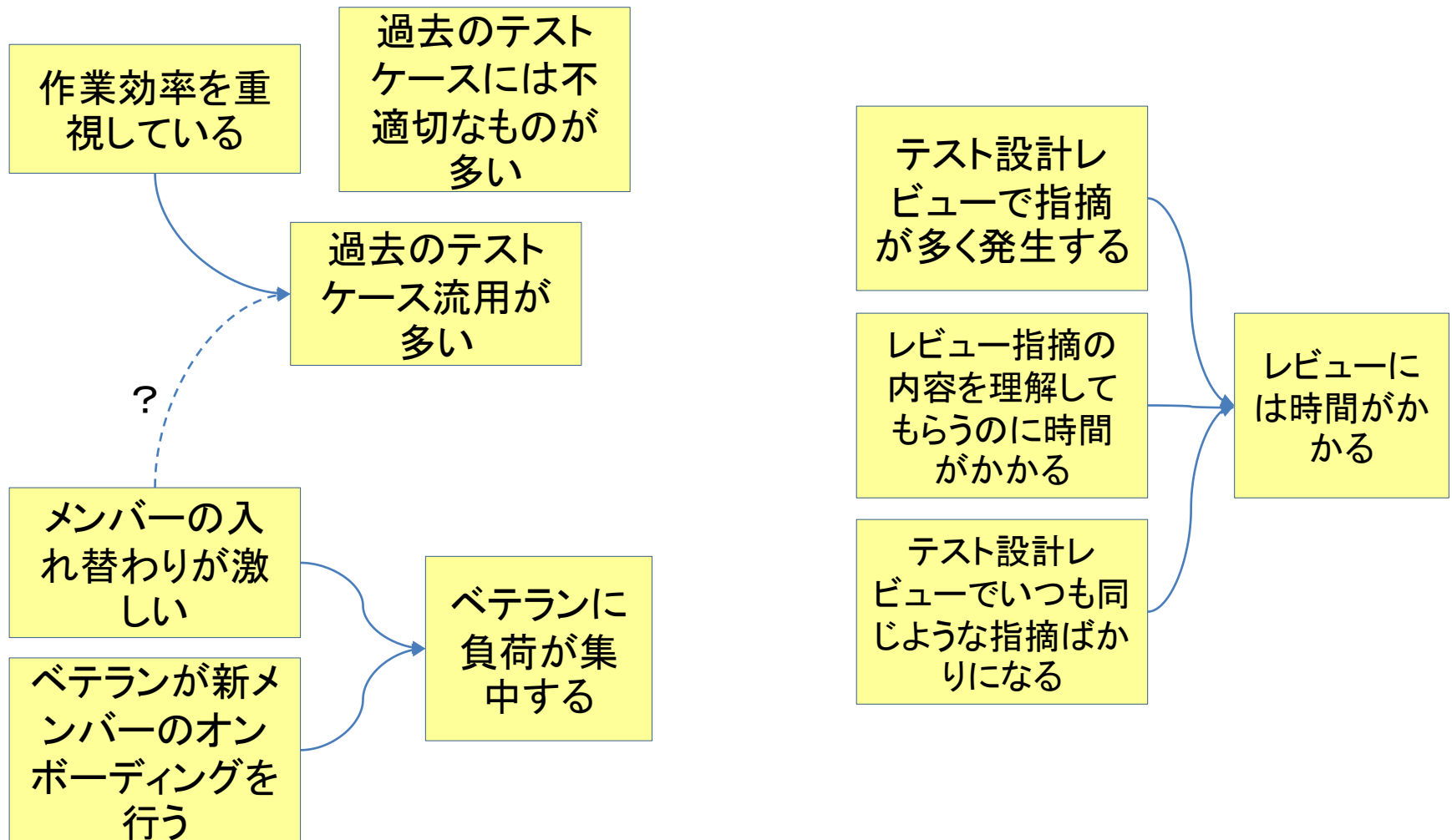
要件・設計・実装・テスト的な要素を配置

例：企画→要求定義→設計→実装→テスト→リリースなど

運営基盤等各種支援的な要素を配置

例：教育・ツール・既存システム・スキルなど

部分構造化～全体構造化へ



STEP3-2: 要素間の因果関係を分析する

要因

結果

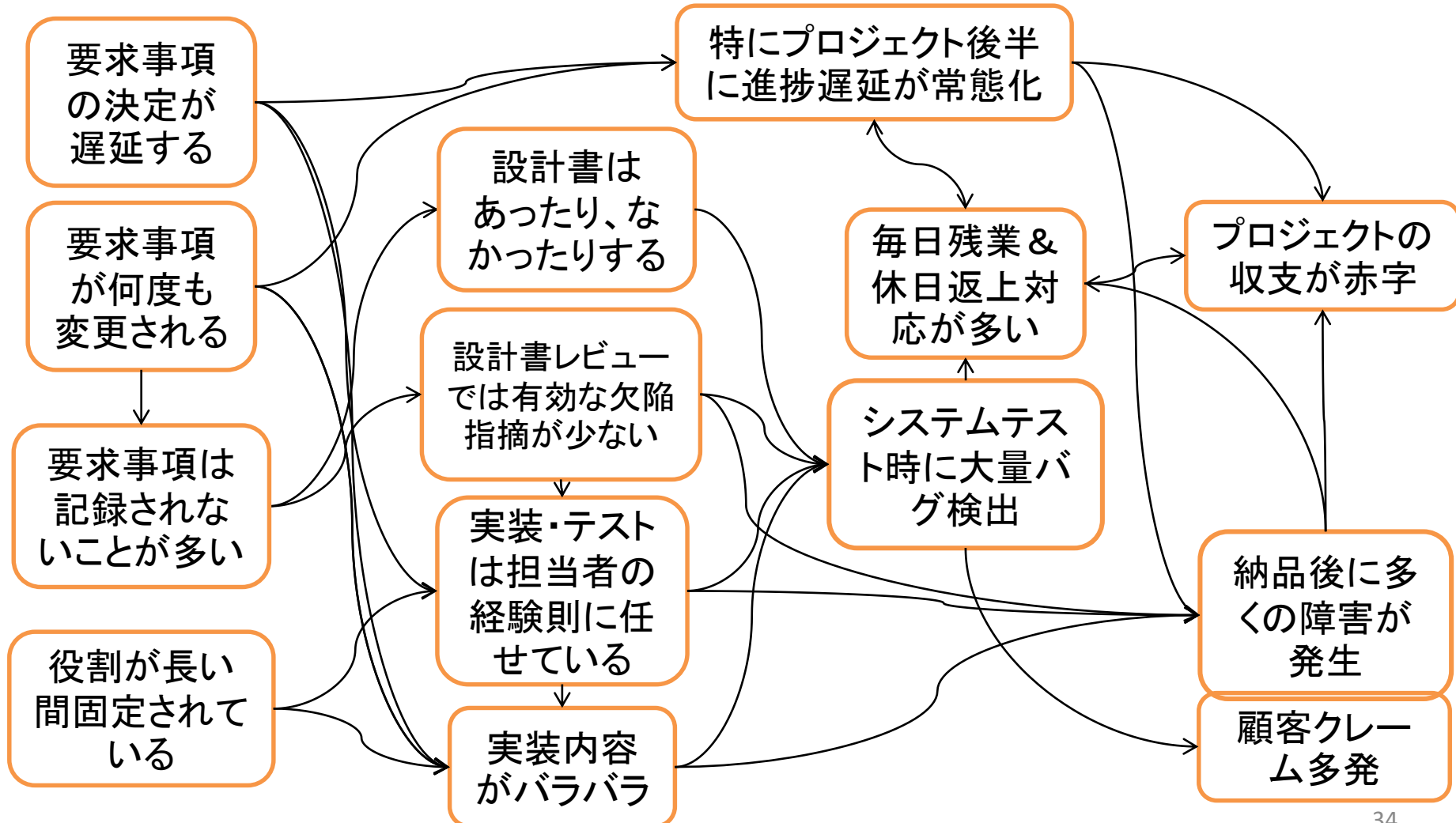
①○○○○○
サービス

②顧客:
□□□□□
▽▽▽▽

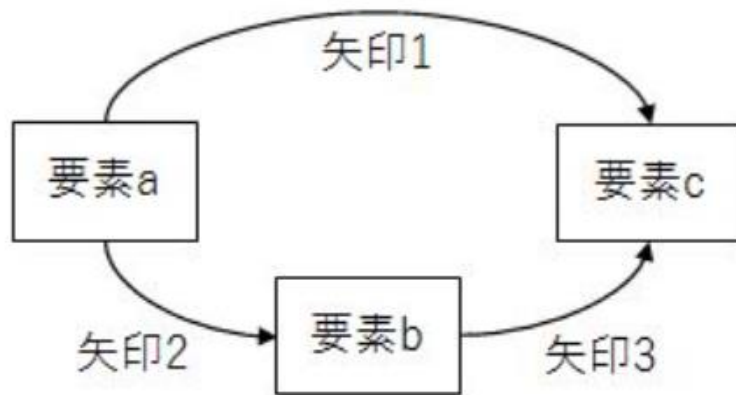
③提供している
価値□□□

④評価方法:
□□□□□
△△△△△

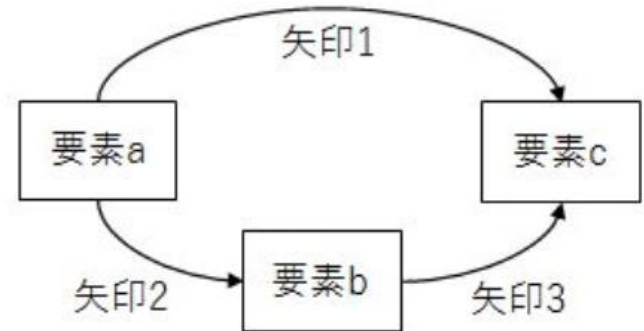
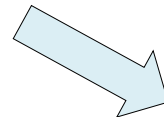
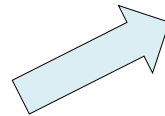
構造化の最中に新しい要素を追加することもある



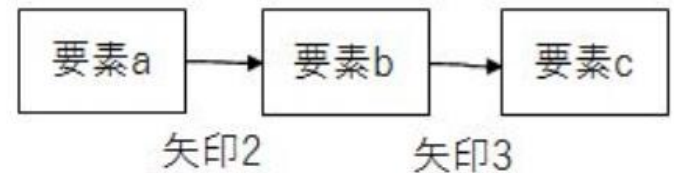
関係線(の意味)を正しくつなぐ



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)

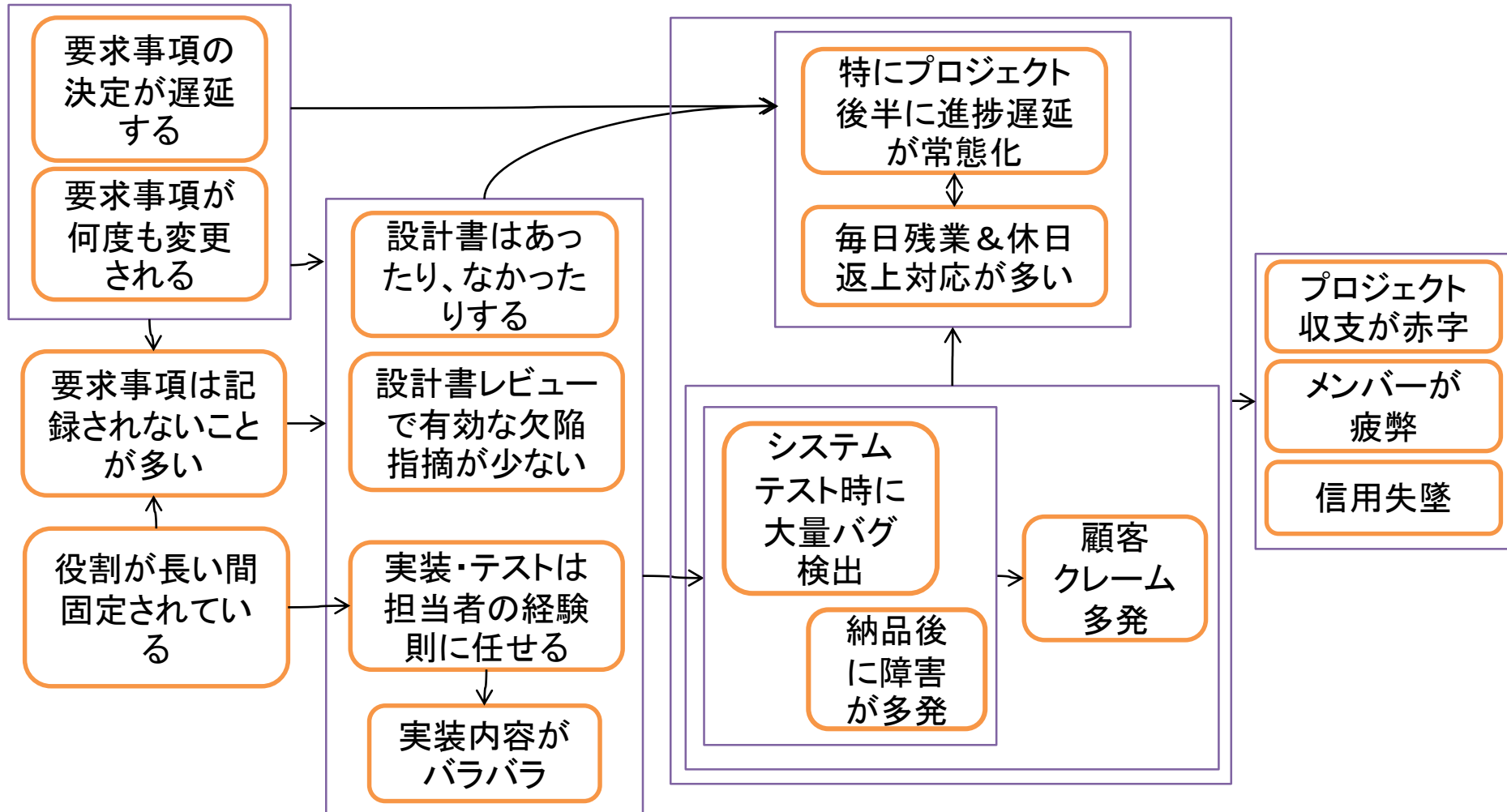
STEP3-3: 要素間の関係や内容を簡潔に整理する

①○○○○○
サービス

②顧客:
□□□□□
▽▽▽▽

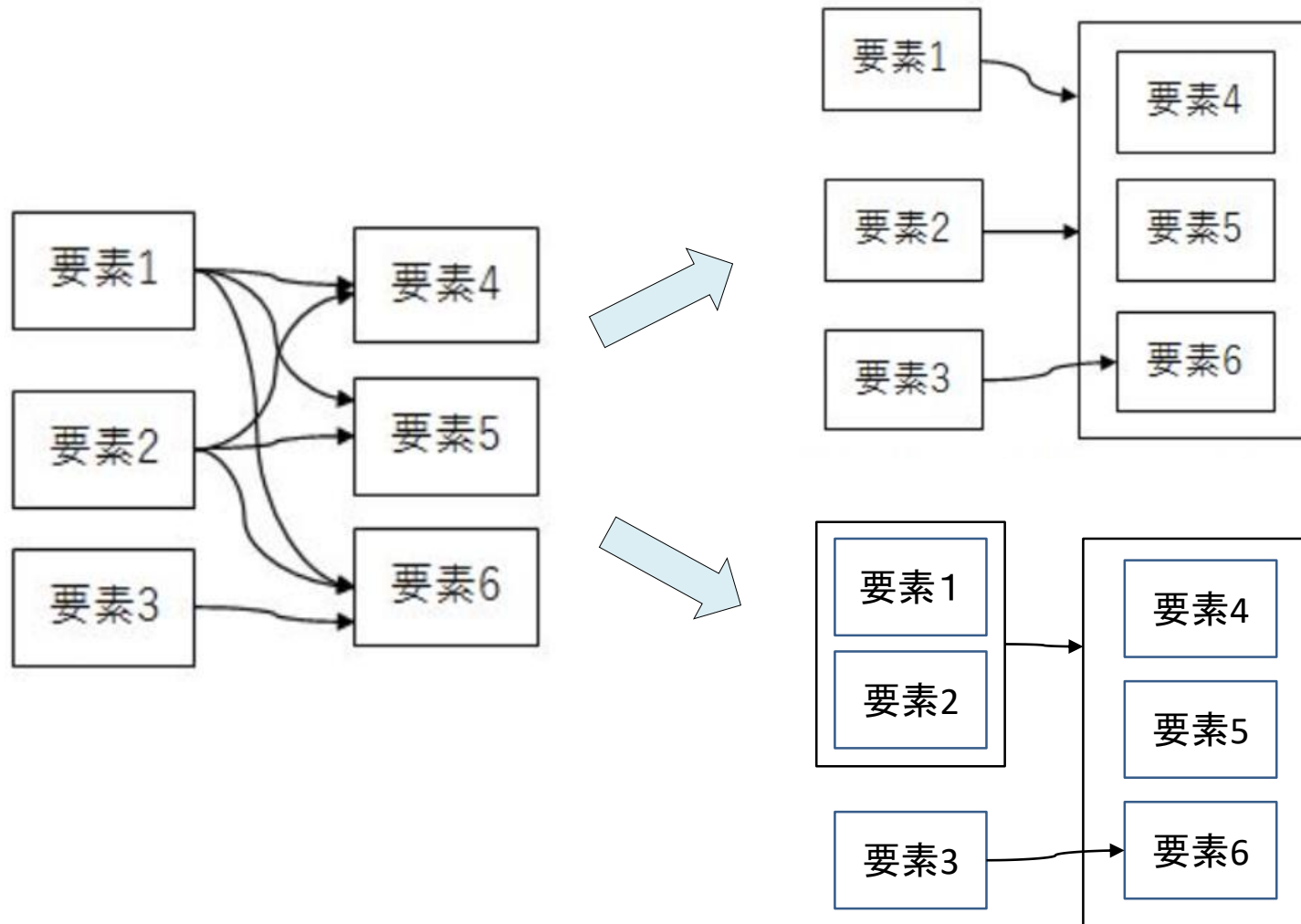
③提供している
価値□□□

④評価方法:
□□□□□
△△△△△



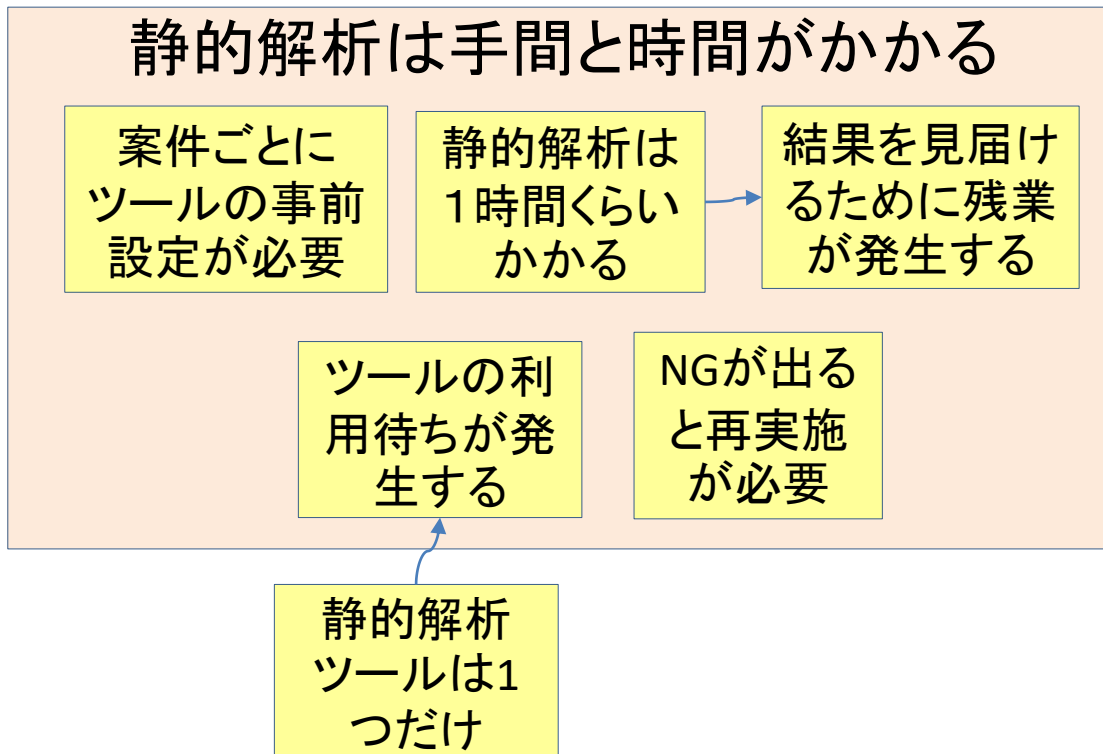
関係線は可能な限り少なく/クロスしないように

関係線が最小・クロスしない＝わかりやすい＝解きやすい



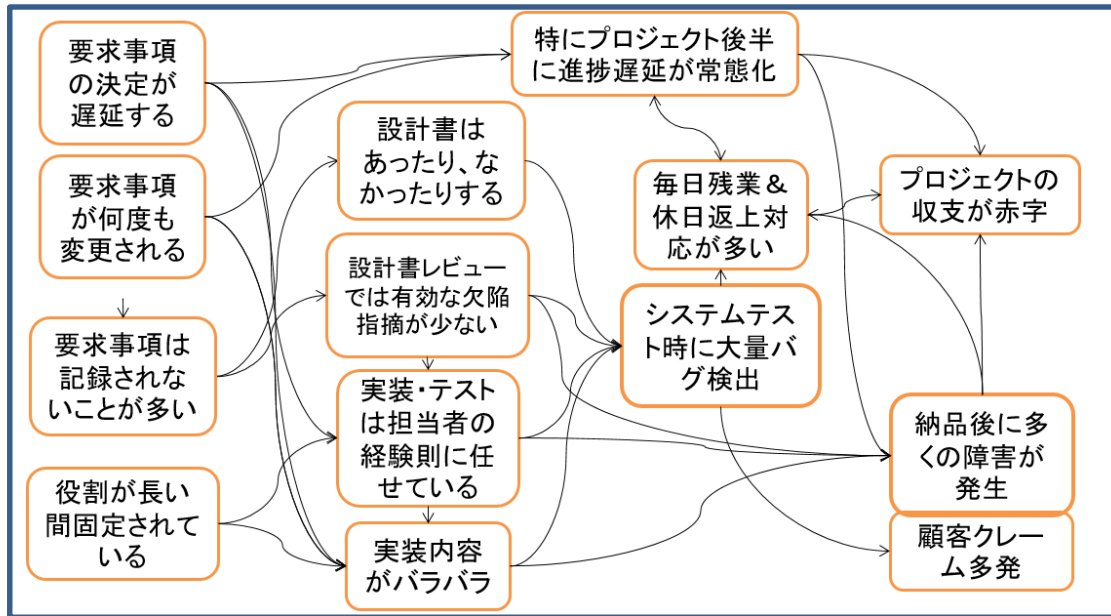
要約で簡潔に

- 要素数が多い場合、類似の要素や関係する要素を集めて要約するのも一案。

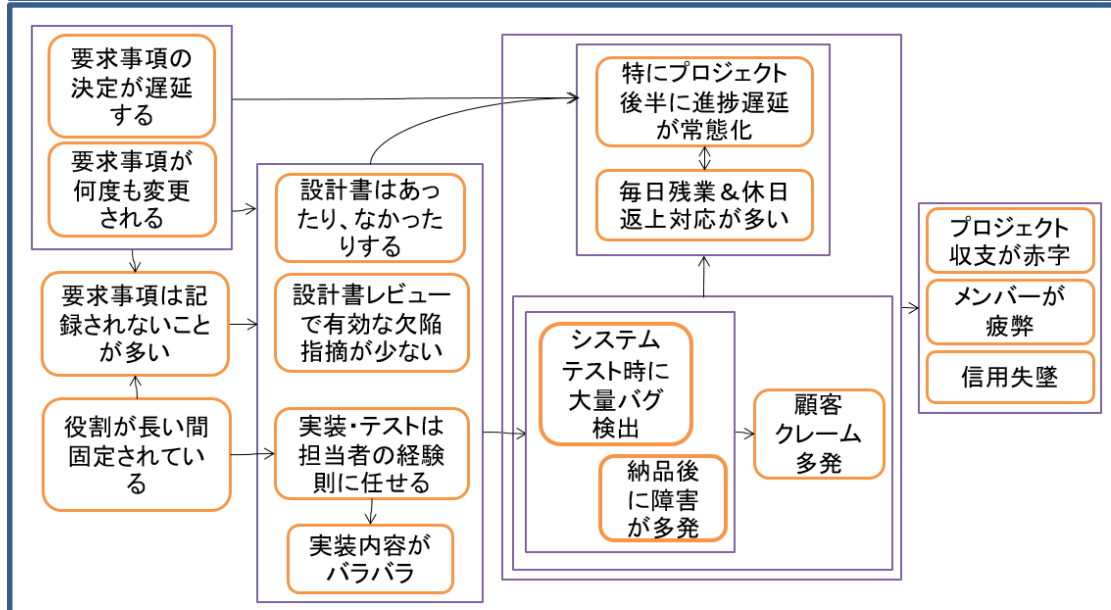


1つの構造図を
「全体俯瞰」しつつ
「詳細把握」する
には、要素数が20
前後であることが
必要。(経験則)

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

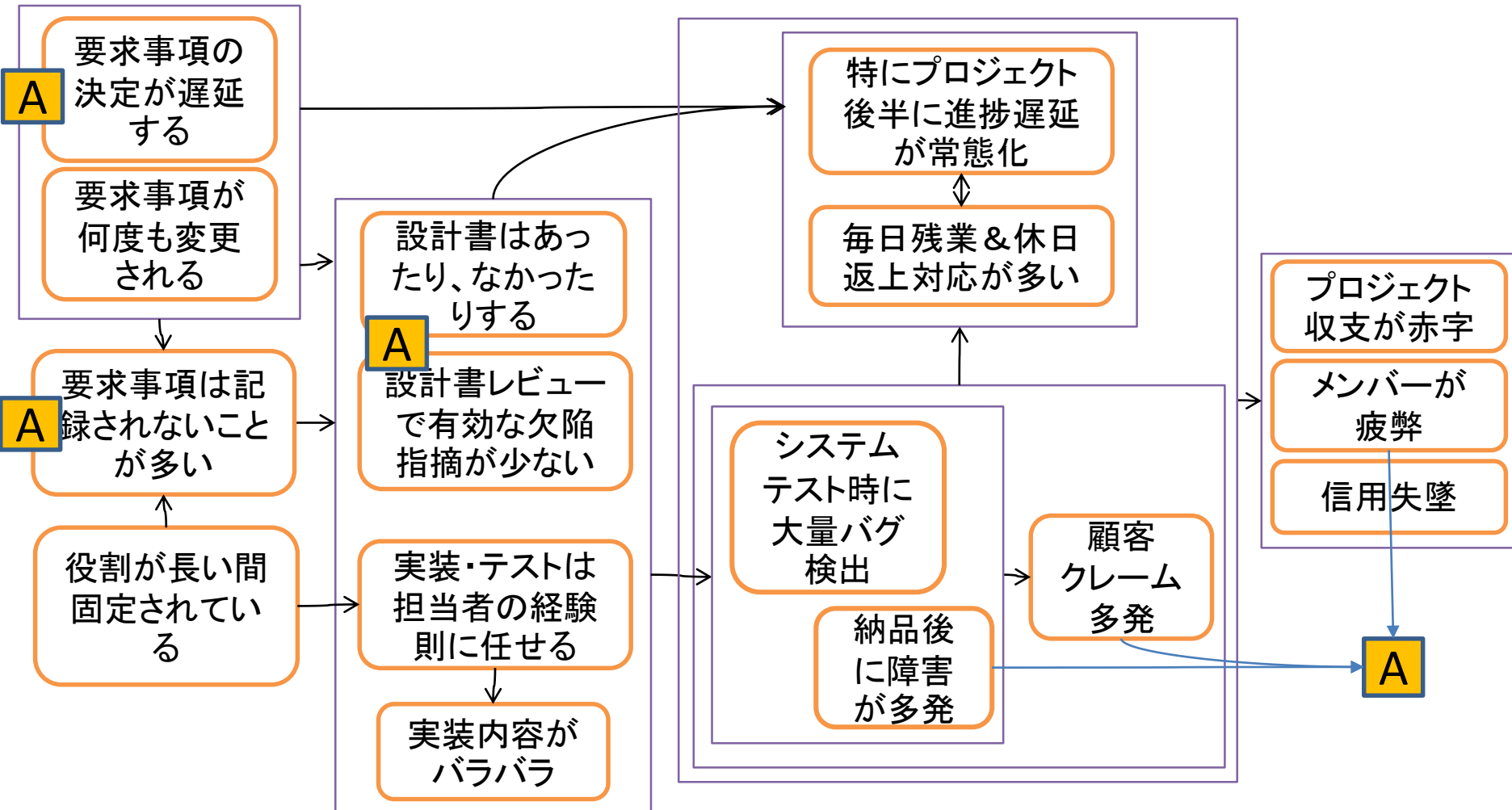
STEP3-4: ライフサイクルで循環する要素を探る

①○○○○○
サービス

②顧客:
□□□□□
▽▽▽▽

③提供している
価値□□□

④評価方法:
□□□□□
△△△△△



STEP3-5: 最終確認

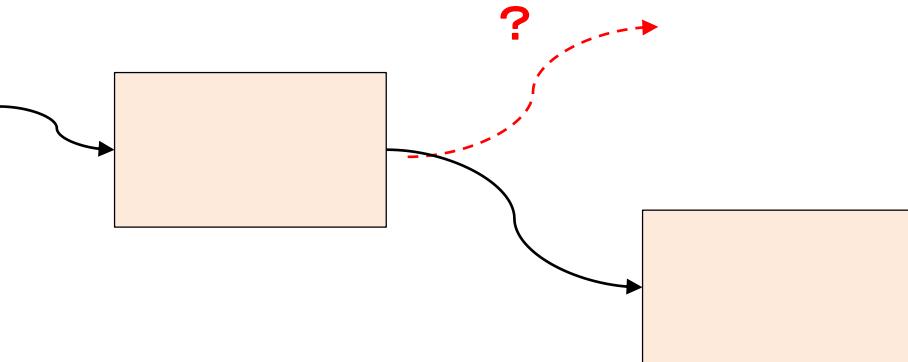
(1) → が刺さって終わっている要素がないか？
(他の要素に影響を与えていないか？)



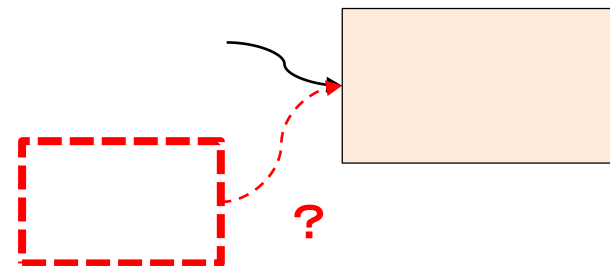
(2) 他の要素に影響を与えているだけの要素はないか？
(他の要素からの影響はないか？)



(3) 他の要素に影響を与えていないか？
(循環を含めて)



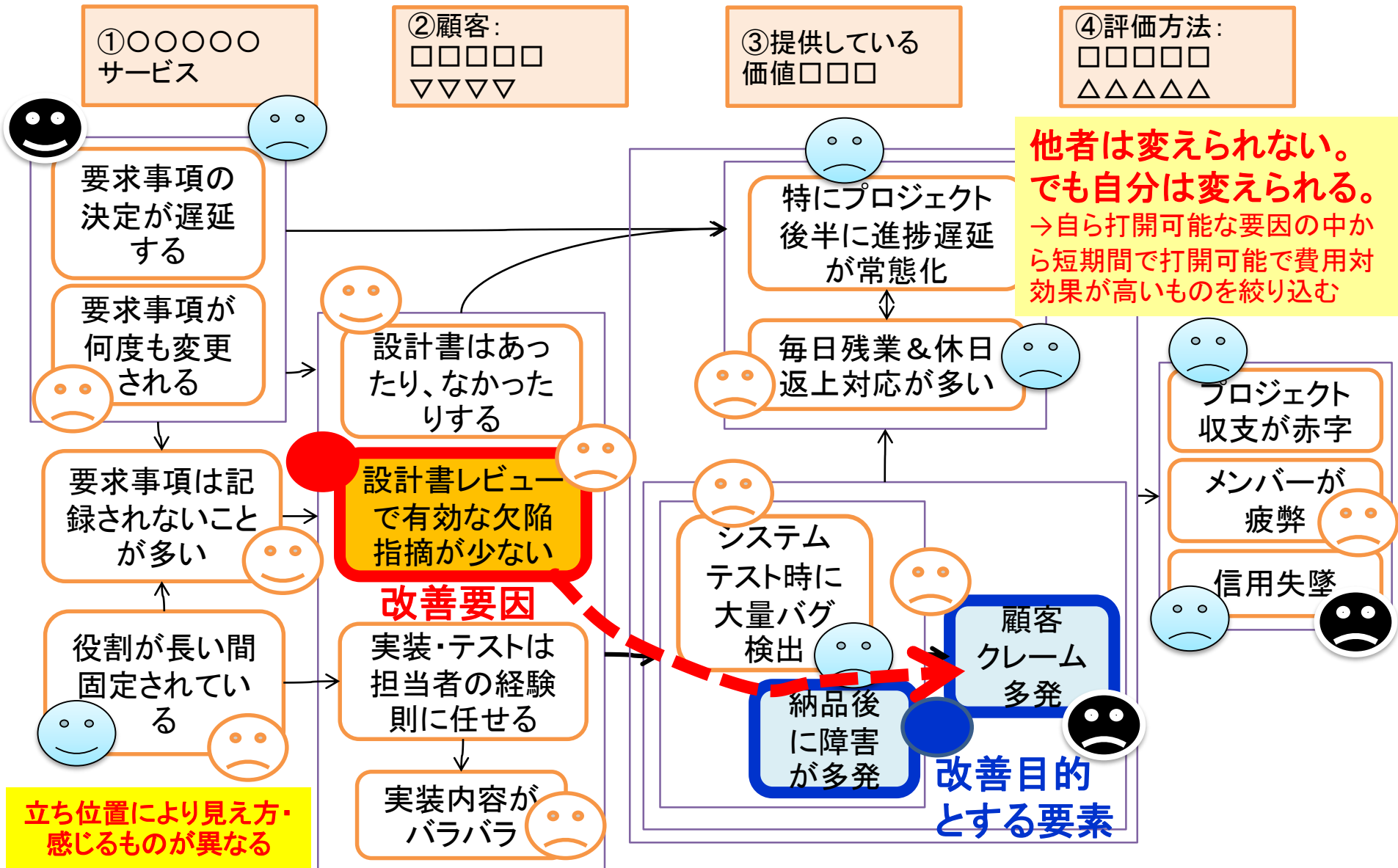
(4) 明らかになっていない他の要素からも影響を受けていないか？



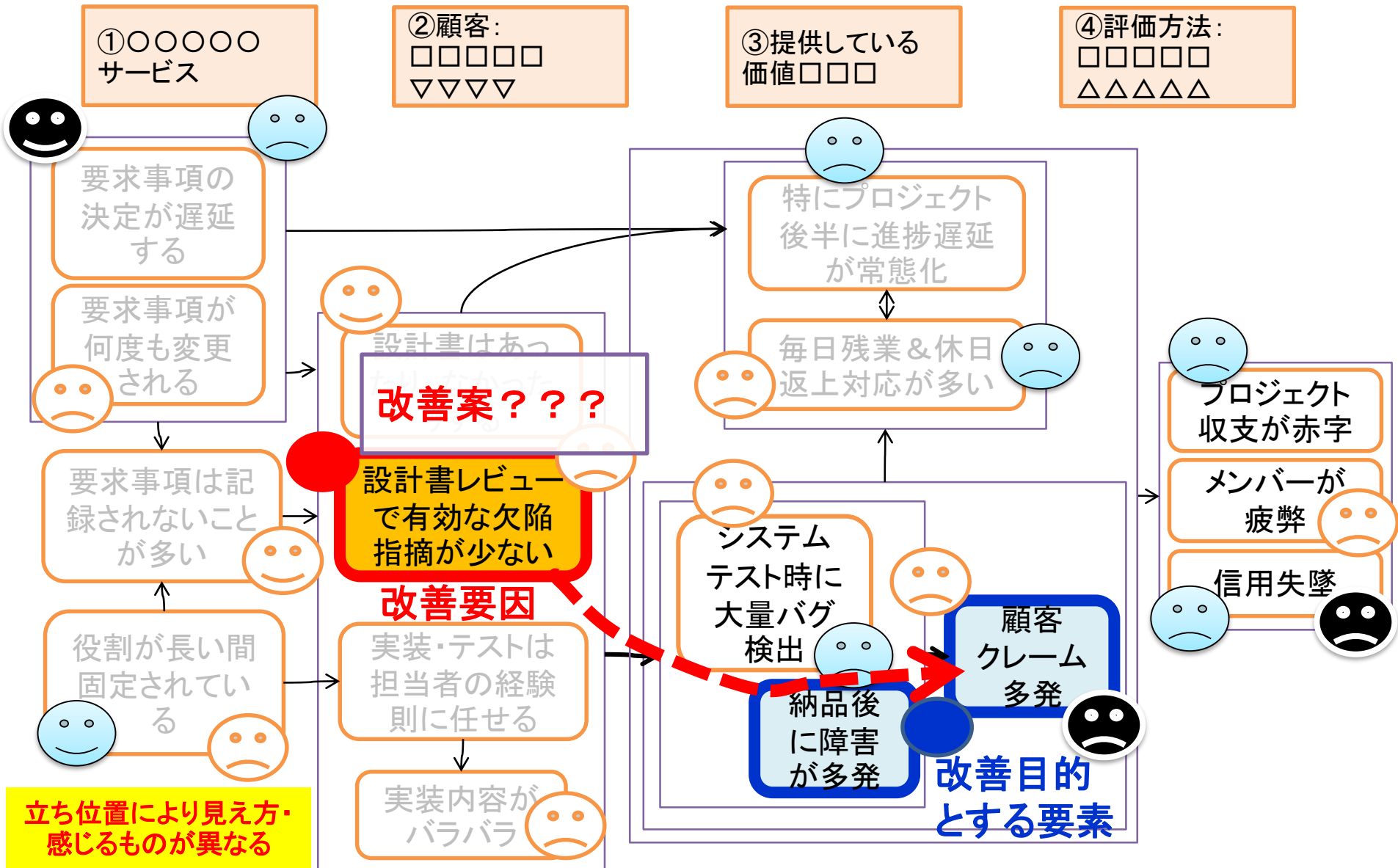
STEP3 ふりかえり結果

Keep	Try
Problem	

STEP4: 改善ターゲット検討・特定



STEP5-1: 改善手段検討・決定



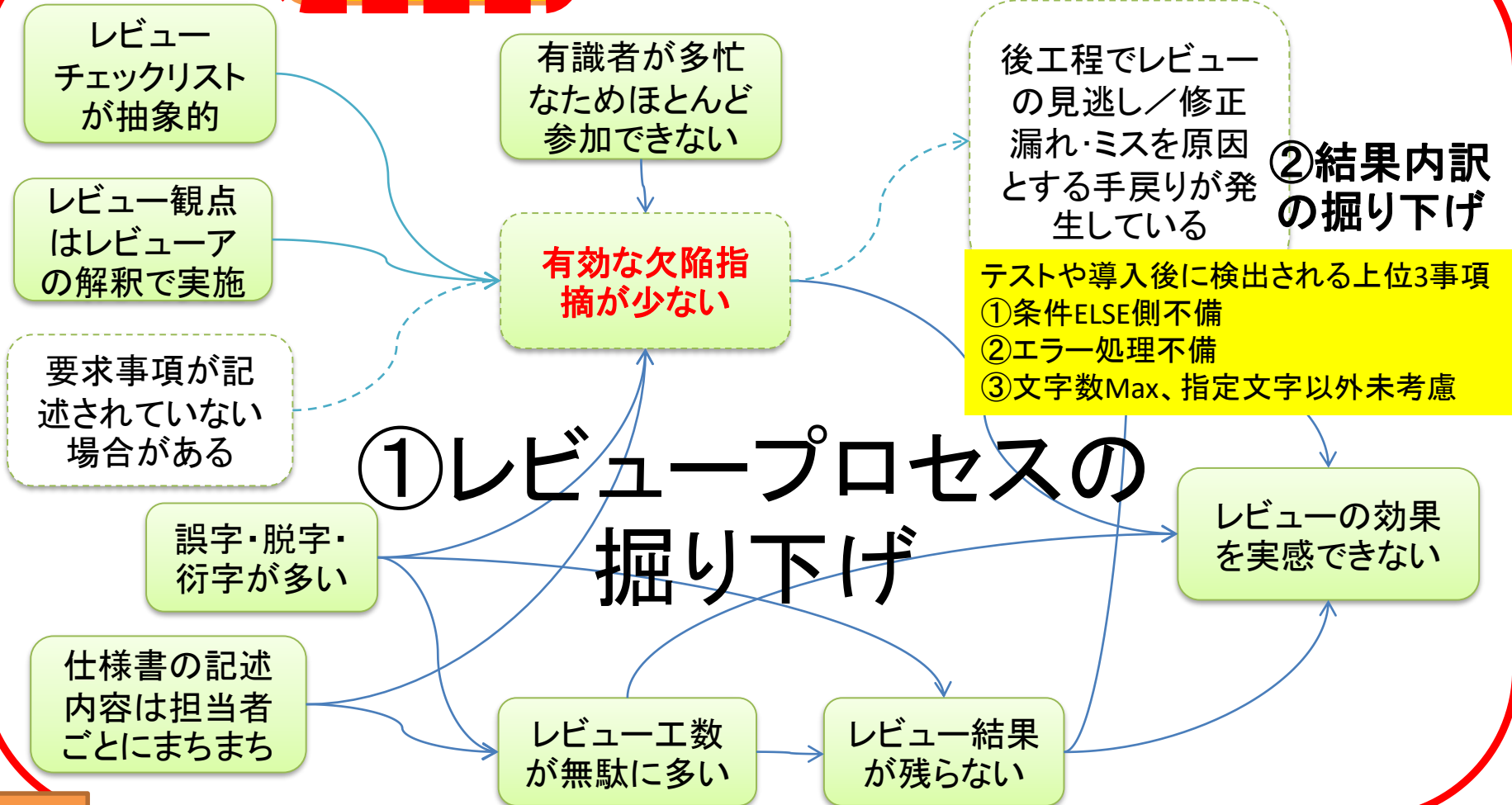
STEP5-2: 改善対象の掘り下げ

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。

上位問題構造図の改善要因

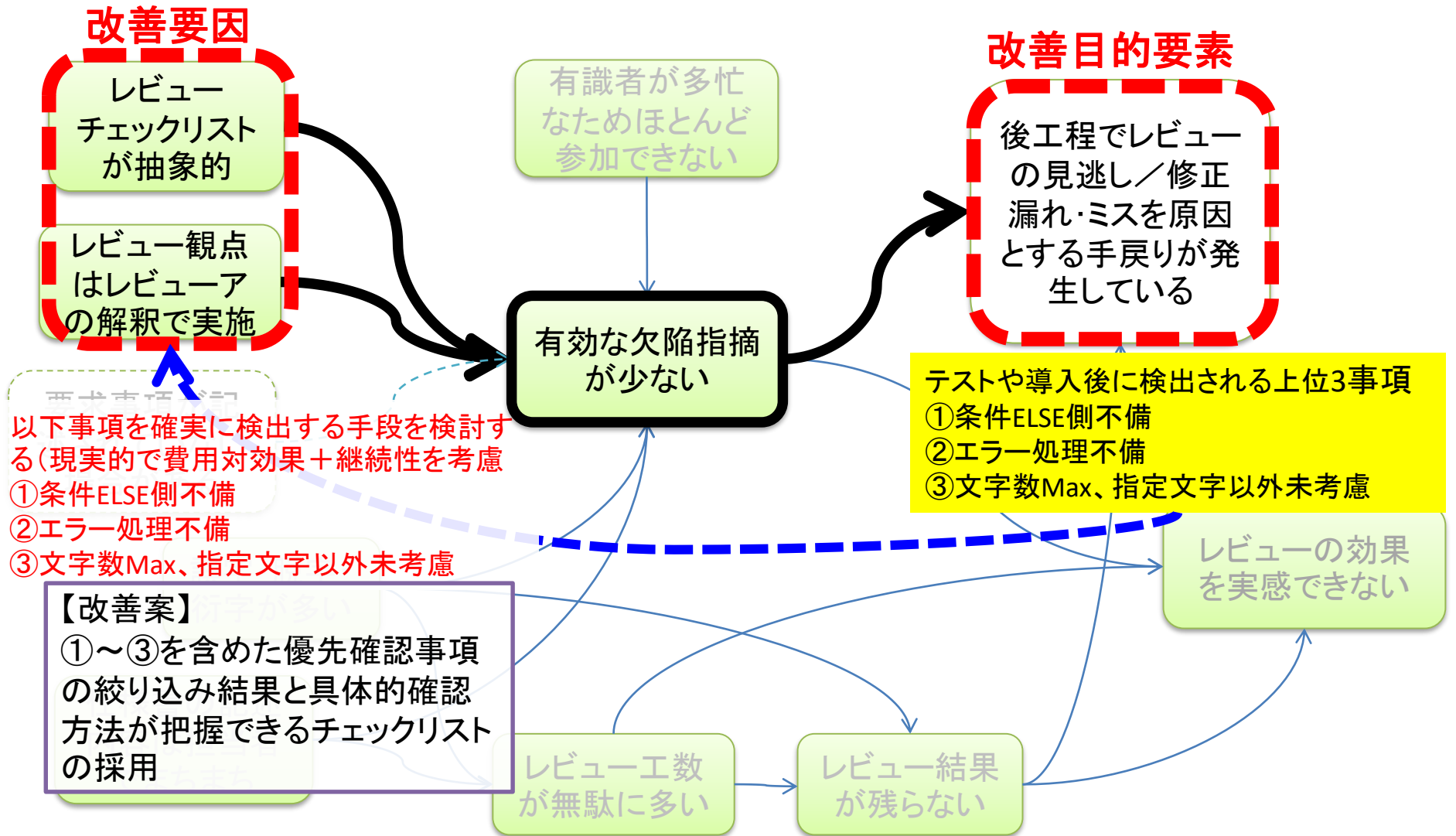
設計書レビューでは有効な欠陥指摘が少ない

改善目的要素の現状



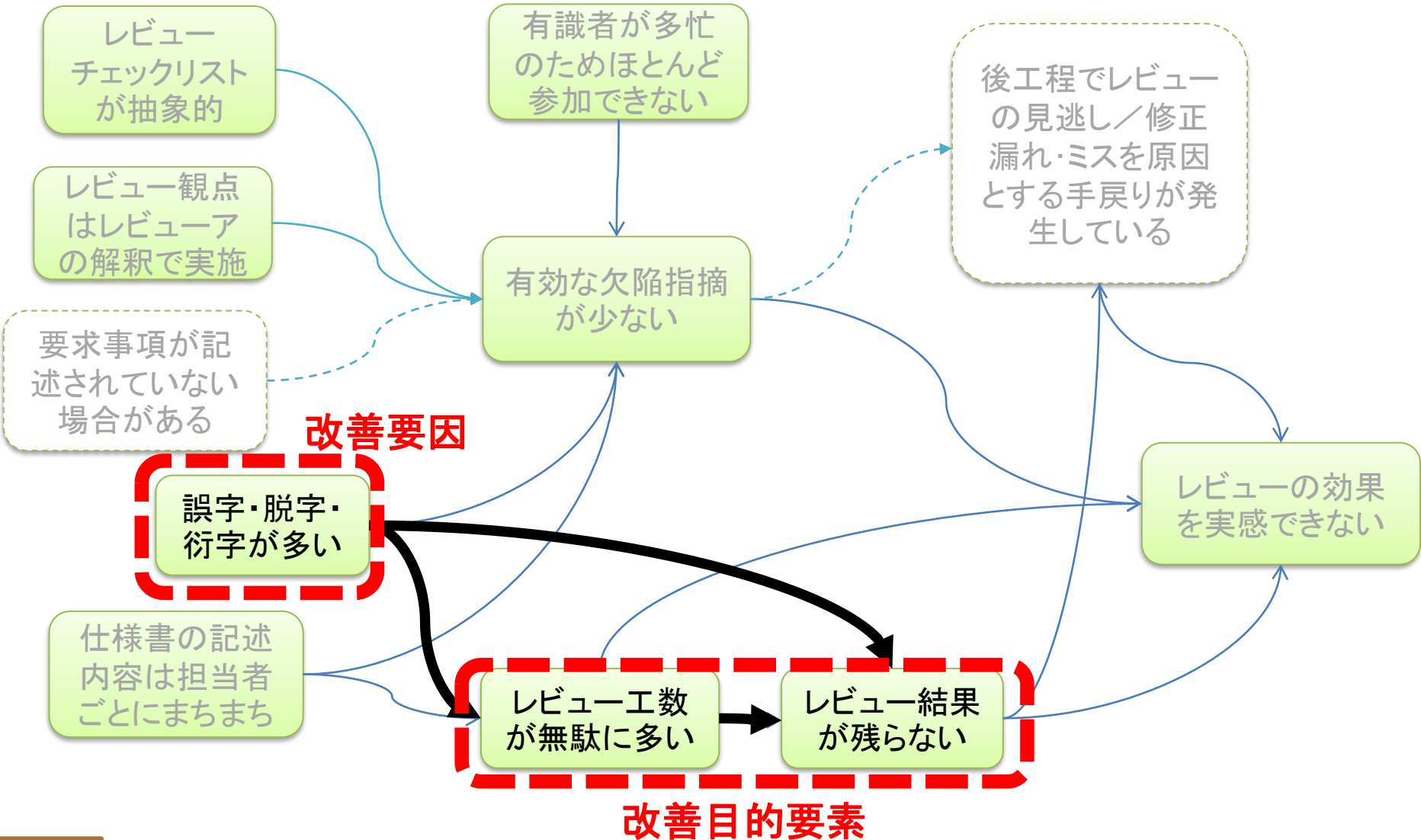
STEP5-3: 改善手段検討・決定

レビューの本質的な改善に着手する例



STEP5-3: 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



ターゲット特定時の考慮事項

- 自ら打開することが可能である。
 - × 打開策実践を誰かに委ねる＝運命を他者に委ねる
 - まずは自ら改善を実践し、効果を上げる。そのあと周囲を巻き込むのがSaPIDの流儀
- 想定する打開策を適用すれば欲しい結果や効果が獲得できる／または、欲しい結果や効果に近づくことができる(可能性が高い)。
 - 効果がないことをいくらやっても無駄
 - いきなり完璧を目指さなくてもよい
- 打開策実践開始～効果確認・完了までのリードタイムが短い。
 - 効果獲得までのリードタイムが長い＝失敗確率が高い
 - 短い時間で効果を確認できる＝改善実践モチベーションが高まる

解決しない



STEP4-5 ふりかえり結果

Keep	Try
Problem	

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

改善要因

設計書レビュー
は実施していな
い場合が多い

テスト時に大量バ
グが検出され、想
定以上の工数と
期間がかかる

施策系改善目標

例1: レビュー実施

例2: レビュー実施率

改善要因

設計書レビュー
は実施していな
い場合が多い

改善目的要素

テスト時に大量バ
グが検出され、想
定以上の工数と
期間がかかる

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

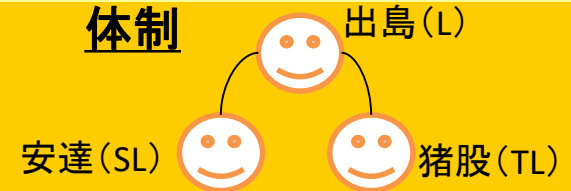
STEP7: 改善計画立案

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

テーマ: ○●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

体制



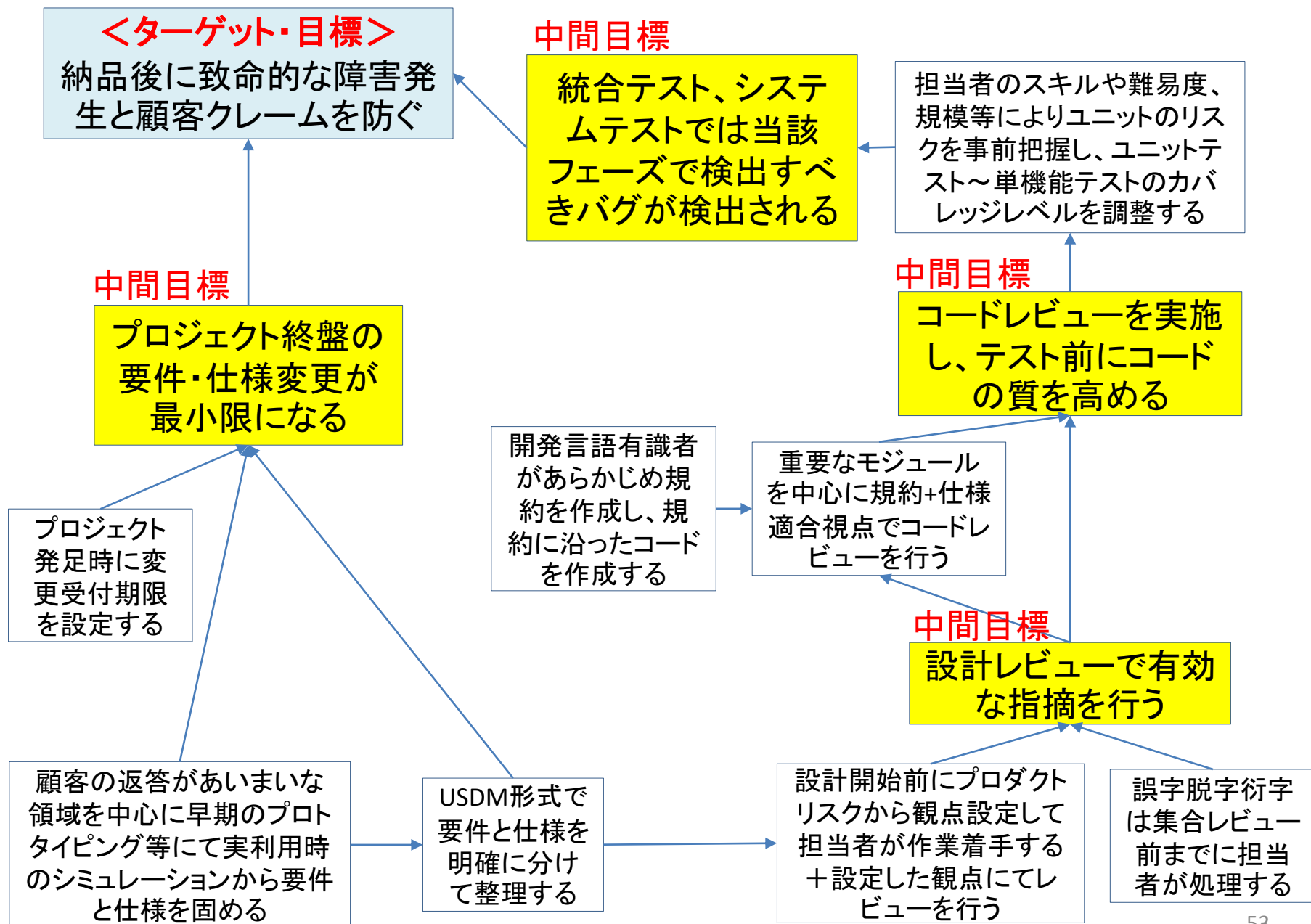
1.業務概要	△▲部の統括運営・管理を行っています。 部署運営の効果・効率に直結する運営の最適化を目指しています。			
2.着目した問題	・内部、外部関係者間で共有、活用する□■情報の作成・修正・共有に手間と時間がかかっている。共有情報が有効活用されていない。部署内で共有情報活用が進まない理由に情報提供のタイムリーさの欠如が存在している。			
3.要因と改善方法 ※参照:5.改善前後の運営イメージ	要因1: 共有情報の作成を元ネタ情報FIX後に手作業で行い、その後再確認、共有、連絡発信の流れになっている。 →改善方法1: 元ネタ情報がFIXするまでの過程で同時並行で作成・関係者確認を行う。 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

アンビシャスターゲットツリー例 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必要な活動
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例 2/2

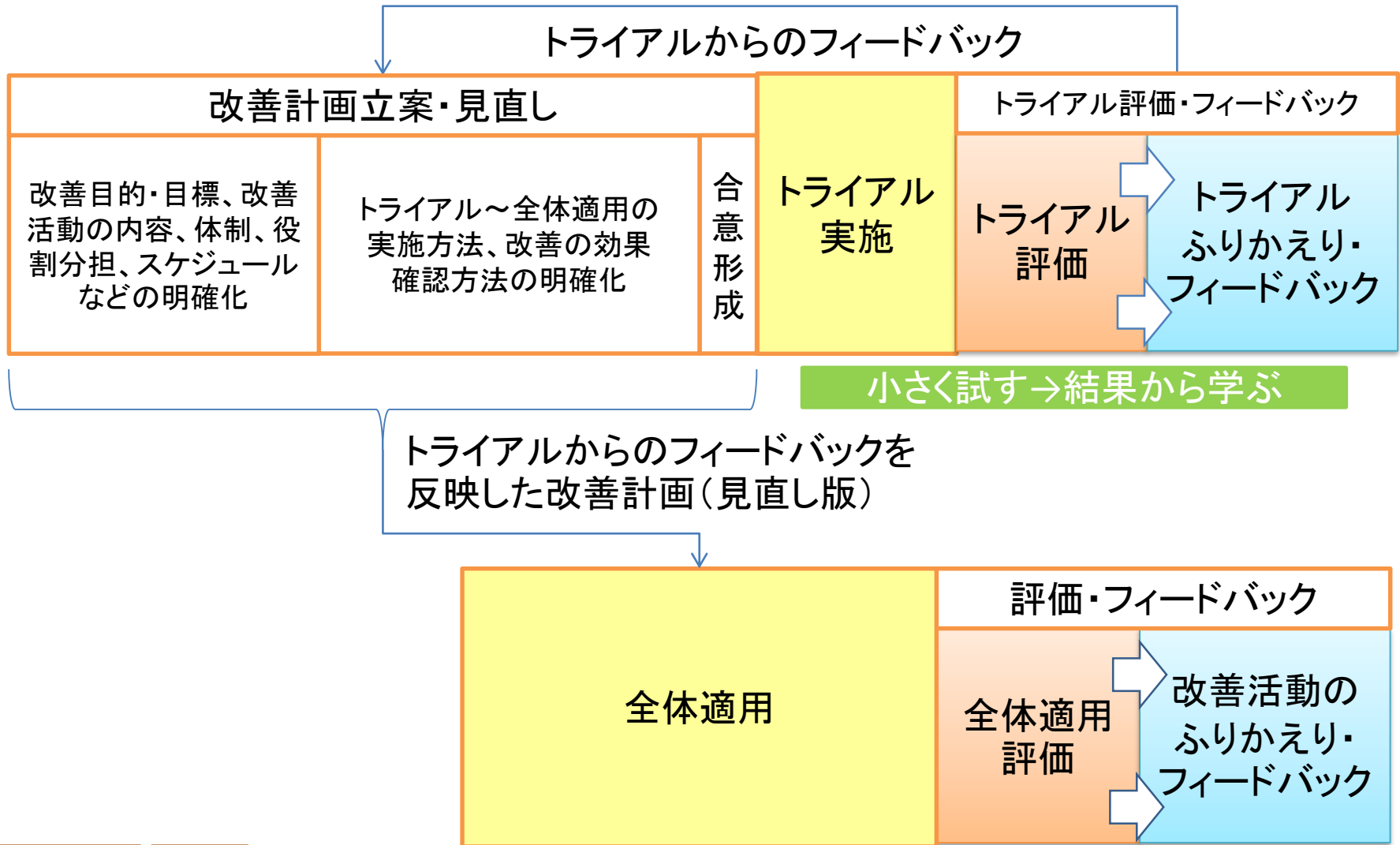


STEP6-7 ふりかえり結果

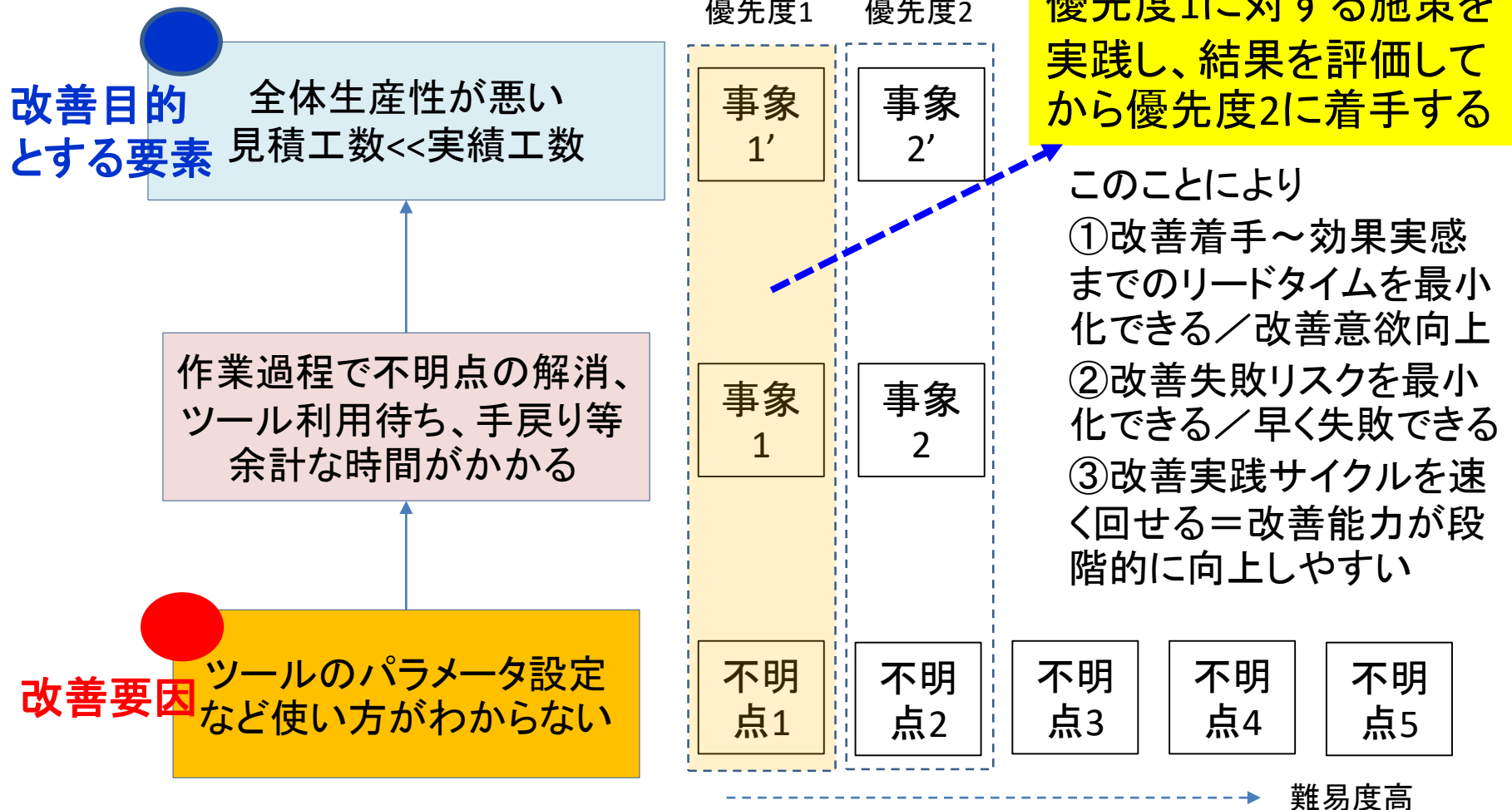
Keep	Try
Problem	

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



ターゲットを分割して優先度を付与 一つずつ完結させる



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

ふりかえり結果例

EP: 良い点・9

まずはよかったこと
からスタート

(俺、ナイスプレー！)

<成果>

A.Oの取り組みに賛同・参加したメンバーとチームという意識が育ち、
のはじめ、(2週間経過後)は **コロナウイルス！**
に毎日のことだが、免疫対策だけではわからぬことが得られる。

ふりかえり結果例

重要事項・改善必要事項→重要事項はTRYで対策

言いにくいこと
でもスバッと

大事なことは強調

分類して まとめる

感じたことも
忘れずに

Copyright © Kenji Adachi. All Rights Reserved.

ふいかえい結果(例)

Keep K
K1: 皆が能動的に動いたので管理能力向上
目標メンバーあたりのリスク識別3項目は全員達成！すごーいぞ！（あ）

その場ですぐ
に次の活動
準備

Problem P 作成者側の言葉で記載されていたが、受け手側で理解できず質問が相次いだ。
(し)

△:受け手でも分かる言葉で表現する
○:成果物レビューチェック欄点Keywordに
“作成者側のローカル言葉”を追加し、レ
ビュー時に確認する

業務プロセスKnow-How/リスト

Process	 リスク対策	 リスク 
管理	H24管理能力向上目標口口	H23能力向上目標全員達成！
設計・実装	受け手が分かる言葉で表現する	作成者側の言葉で記載されていたため、受け手側で理解できず質問が相次いだ。
Review	Keyword12 “作成者側のローカル言葉”	

次の活動時にそのまま使う

イロハニホトテ一ツクニ

(4)SaPIDの適用

SaPID+適用：自律運営への5段階

【MODE1】毎日+週次のふりかえりによりチームで問題発見・解決を実践する



【MODE2】個別改善が自然と身につき、あたり前にする



【MODE3】問題モデリングを高度化しながら本格的改善にも取り組む



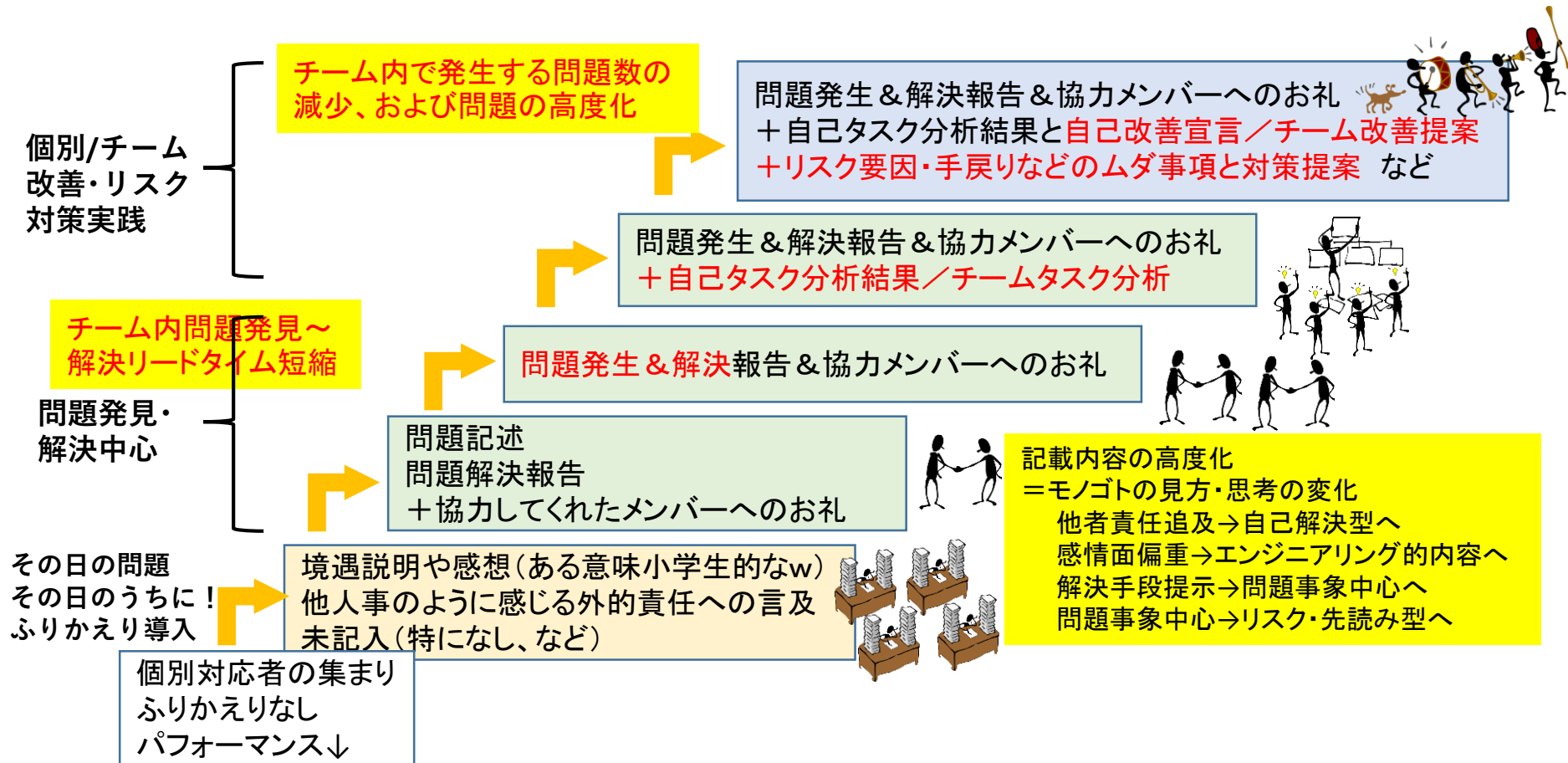
【MODE4】予防処置をも実践し、最終目標を定め、徐々に規模・難易度を高めながら取り組む



【MODE5】総合的なリスクマネジメント実践による自律運営の実現

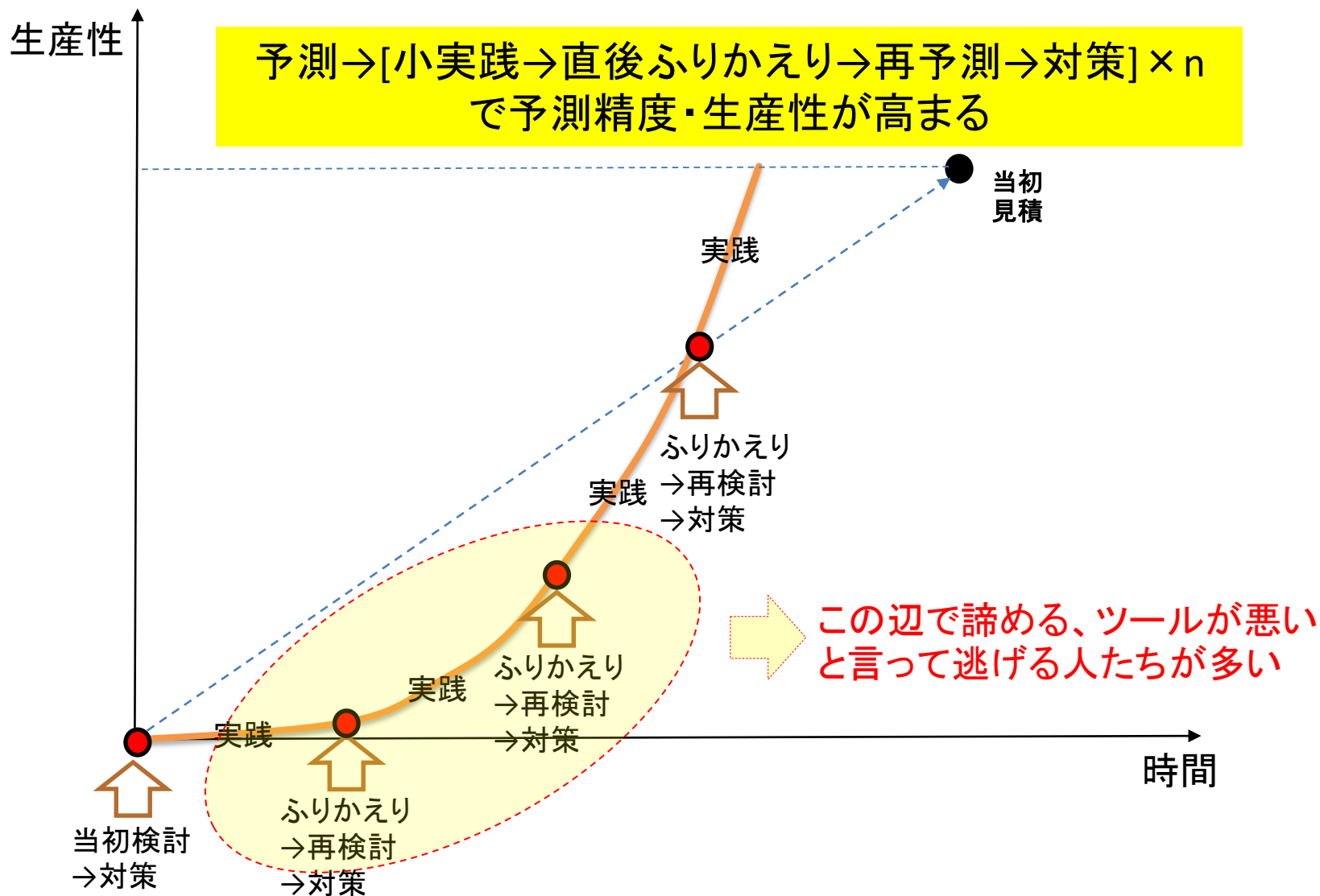
あるチームのふりかえり内容の変化例

毎日個人ふりかえり→チーム内共有・対応／週1回集合ふりかえり



SPI Japan2015投稿「自律型プロジェクトチームへの変革アプローチ事例
～チームの価値観変容を重視し、問題モデリングを活用したSaPID流プロセス改善アプローチ～より

適用してから効果が出るまで



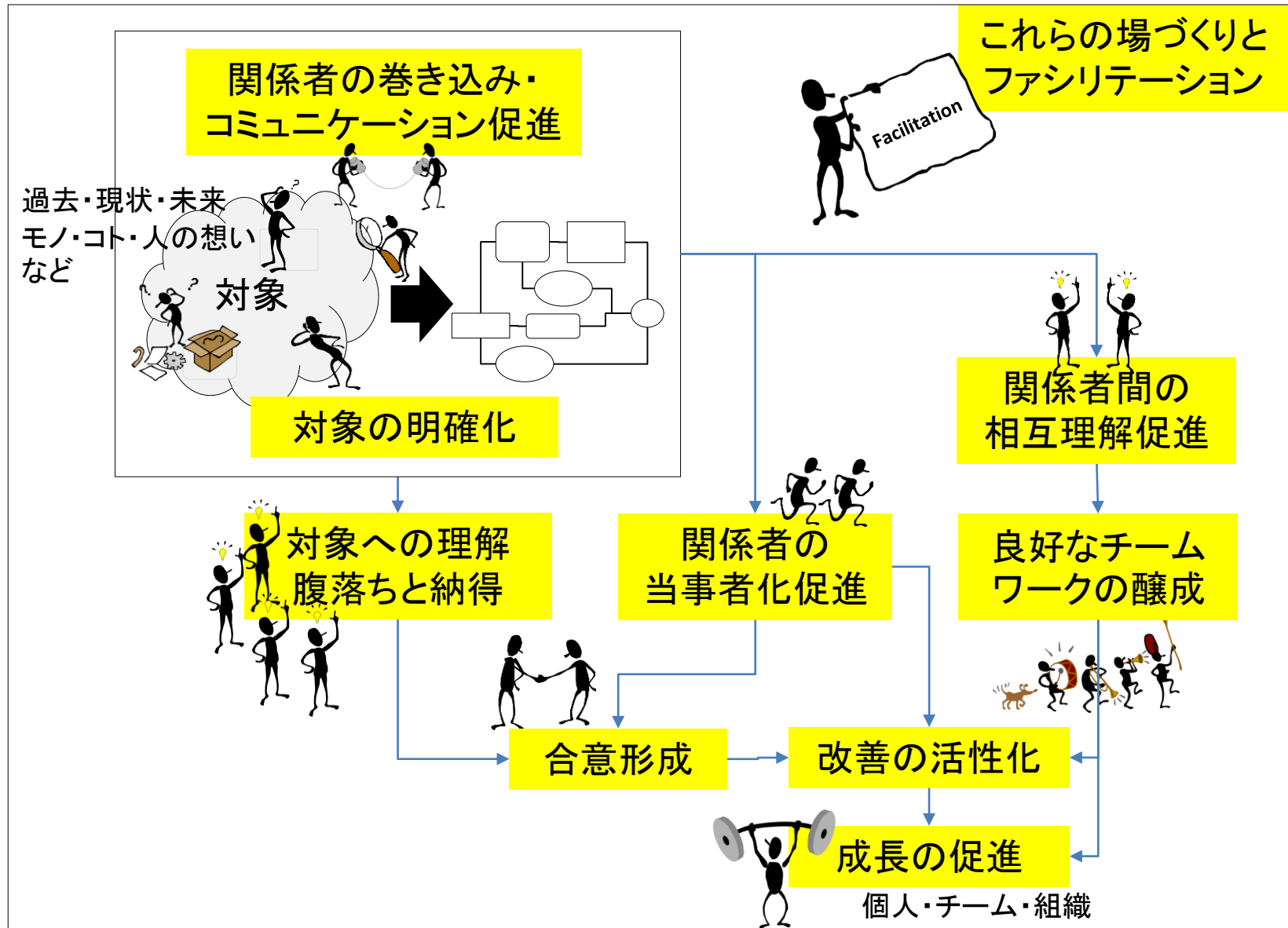
適用上の注意(続き)

- [計画→実践→ふりかえり]のサイクルタイム
- 当手法を無効化する要因
 - 問題を共有できない文化
 - 圧の強い人、失敗への辛辣な批判が得意な人
 - 自分のことしか考えない自己中心的メンバー
 - マイクロマネジメント実践者
 - 不適切な介入(上級マネジメント) などなど

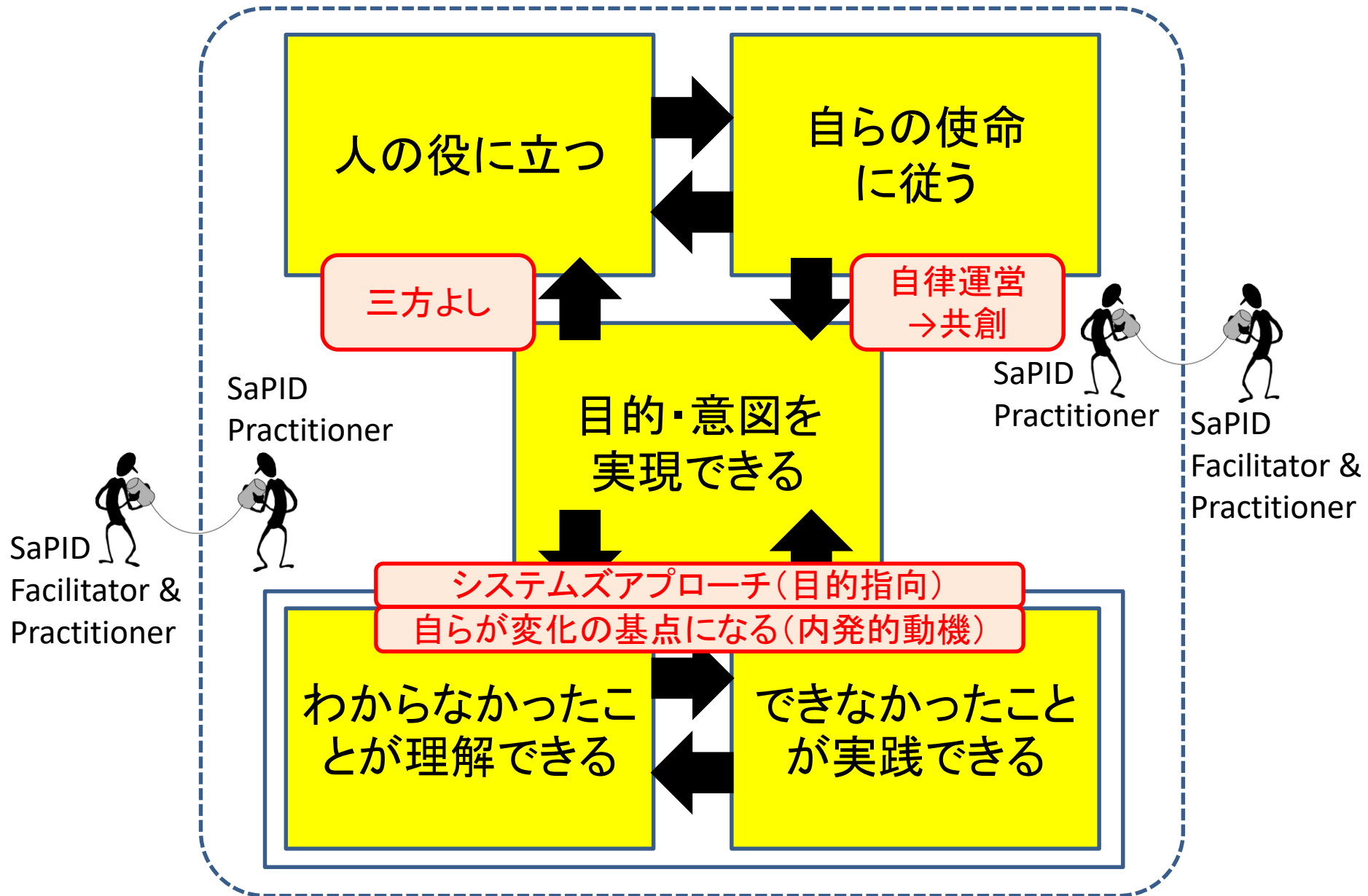
短い方がよい

人のやる気を削ぐ、チームワークを無効化するのは意外と簡単w

SaPIDでできること・やっていること



(依存)→自律→共創(相互依存)による幸せ実現モデル



SaPIDファシリテーターが実践を支援



- SaPIDファシリテーターの役割＝SaPID実践の伴走者
 - 管理者や技術者が、管理者や技術者が自らの背景や経緯、さまざまな事情や境遇を理解し、自ら前に進む答えを出してマネージャーライフ、エンジニアライフを発見的に歩んでいくのを支援すること。

資格分類	資格名	概要
SaPID Master (マスター)	SaPID Master	SaPIDをさまざまな状況に適応し、より効果的に発展させることができる、SaPIDの指導者。 ・SaPID(MODE1～5、およびSTEP0～9を含む)の理解に基づく実践・支援が可能で、SaPID Facilitator、SaPID Practitionerを育成することができる ・SaPIDの強み、弱み、他の方法論や手法との関係性を把握し、SaPIDをより効果的にバージョンアップすることができる
	SaPID Facilitator (ファシリテータ)	適切な理解に基づくSaPID実践支援(伴走)者 ・併せて自らが所属しない個人、チーム、組織のSaPID適用を支援し、最低限STEP1～9を実践・完了した実績が複数ある。
SaPID Practitioner (実践者)	SaPID PreFacilitator	SaPID Facilitatorの役割とあり方を理解している方。
	SaPID Practitioner	適切な理解に基づくSaPIDの実践者。
	SaPID PrePractitioner	SaPIDの概要と詳細を理解している方。

(5) (参考) SaPID実践事例

これまでの関連事例発表

[SPI Japan 2007](#): 現場の様々な事実情報分析に基づく現実的な改善アプローチのご紹介－システムズアプローチを活用した改善実践事例－

<http://www.jaspic.org/event/2007/SpiJapan/2A3.pdf>

[SPI Japan 2011](#): ふりかえり実践方法の変遷による業務運営プロセスと成果の改善

http://www.jaspic.org/event/2011/SPIJapan/session3B/3B4_ID008.pdf

[SPI Japan 2012](#): システムズアプローチによる問題発生構造分析とPFD (Process Flow Diagram) を用いたプロセス改善

http://www.jaspic.org/event/2012/SPIJapan/session3A/3A3_ID009.pdf

[SPI Japan 2012](#) 【最優秀賞受賞】: システムズアプローチに基づくプロセス改善メソッド: SaPID が意図するコト～プロセスモデルをより有効活用するために／そして現場の自律改善運営を促進するために～

http://www.jaspic.org/event/2012/SPIJapan/session3A/3A4_ID023.pdf

[SPI Japan 2013](#) 【実行委員長賞受賞】: SaPID 実践事例より～改善推進役がやるべきこと／やってはいけないこと 現場が自らの一步を踏み出すために

http://www.jaspic.org/event/2013/SPIJapan/session2B/2B3_ID011.pdf

[SS2013](#) 【最優秀発表賞受賞】: 「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://sea.jp/ss2013/accepted_papers.html#s3

[派生開発カンファレンス2013](#): 問題構造分析とPFDの併用による現実的・段階的な改善実践方法の提案～PFDを使いこなす能力を確実に身に着けるために

http://affordd.jp/conference2013/xddp2013_p8.pdf

[SQiP2014](#): 「プロジェクト運営と改善実践の連携・一体化」～プロジェクトマネジメントにおけるSaPIDシステムズアプローチ活用事例

https://www.juse.jp/sqip/symposium/archive/2014/day1/files/happyou_B1-3.pdf

SPI Japan2015 【わくわく賞受賞】:「自律型プロジェクトチームへの変革アプローチ事例」～チームの価値観変容を重視し、問題モデリングを活用したSaPID流プロセス改善アプローチ～

http://www.jaspic.org/event/2015/SPIJapan/session3C/3C-3_ID012.pdf

JaSST2016東京 【ベストスピーカー賞受賞】:「レビューの目的・観点設定の効果と課題」

Slideshare: <http://www.slideshare.net/AdachiKenji/ss-59510938>

SPI Japan2017 【特別賞受賞】:「自分事化影響要因に着目した中期経営計画立案・展開への共創アプローチ[現状分析～計画立案編]」

http://www.jaspic.org/events/sj/spi_japan_2017/

JaSST2018東京 【ベストスピーカー賞受賞】:「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」

<http://bit.ly/2FLtpHU>

SS2018札幌 「リスク構造化を用いたリスクマネジメント手法の提案と効果分析」
～「未来予想図」を用いたリスクマネジメントPDCAサイクル～

<https://www.slideshare.net/AdachiKenji/ss2018-sapidtocpresentation>

JaSST2019北陸 招待講演 「そのレビュー、大丈夫ですか？ ～現状レビューの問題発見・解決」

<http://www.jasst.jp/symposium/jasst19hokuriku/pdf/S2-1.pdf>

<http://www.jasst.jp/symposium/jasst19hokuriku/pdf/S2-2.pdf>

SPI Japan2019 【特別賞受賞】: 静的×動的プロセス改善の実践と課題 共通性×相違性～見つけ方とつなぎ方

http://www.jaspic.org/event/2019/SPIJapan/session2A/2A2_ID008.pdf

＜最近のSaPID関連の主な事例発表＞

SPI Japan2020 事例発表

FRI(Factor-Risk-Influence)モデルによるリスク構造の見える化

http://www.jaspic.org/event/2020/SPIjapan/session6/6-3_ID008.pdf

価値開発研究会 事例発表

As-is → To-be 図書館システムによる運営の効果と価値の可視化事例

<https://speakerdeck.com/kitanosirokuma/tu-shu-guan-sisutemuniyoruyun-ying-falsexiao-guo-tojia-zhi-falseke-shi-hua-shi-li>

現状分析→価値開発→仕様化 & テスト設計の展開事例解説

第1回 課題認識→必要最低限度のAs-is可視化(SaPID-RDRA As-is分析)

<https://onl.tw/ZtqBDwQ> (概説)

<https://onl.tw/puc2GkR> (SaPIDによるAsIs業務・事業現状分析)

第2回 As-is課題特定(SaPID As-is分析)→価値開発(SaPID To-be開発+RDRA To-be要件定義)

<https://sapidcasestudy.connpass.com/event/229689/>

<https://onl.tw/gq9b9P3> (SaPIDによるToBe業務・事業開発～価値設計)