

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

株式会社HBA 安達賢二・猪股宏史

adachi@hba.co.jp ▪ inomata@hba.co.jp

■発表者

安達 賢二(あだち けんじ)
株式会社HBA



札幌(HBA本社・システムビル)



東京支社



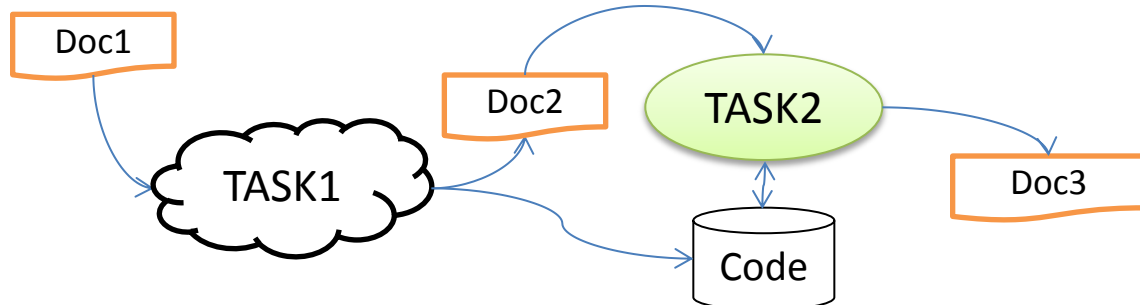
関西ソリューションセンター

経営管理本部 Quality Solution事業 Expert(部長職) H24.4～現職
～H24.3末まで QMS・ISMS/PMS・EMS統括管理責任者
各種プロセス評価・改善推進、管理者・技術者育成・支援担当
全社員参画生産革新活動“スリーム”推進リーダー・テクニカルコーディネータ
およびQuality Solution事業:組織Quality向上を支援するコンサルタントを兼務

■所属学会・協会／外部活動など

1999～ソフトウェア品質関連情報サイトADAMOS(現Software Quality.com)運営
2006～ソフトウェアテストシンポジウム(JaSST)北海道実行委員長(4年間)・実行委員
2006～NPO法人 ソフトウェアテスト技術振興協会(ASTER)会員(2009～理事)、
Test Engineer's Forum北海道(TEF道)お世話係
2007～2013.3 IPA-SECプロセス改善NPT1／NPT6委員
2008～2011 ソフトウェアテストワークショップ(WACATE)実行委員
2009～2010 ソフトウェア技術者ネットワーク(S-open)幹事
2010～ISO/IEC JTC1 SC7 WG24 Very Small Entities(VSE)エキスパート
2011～テスト技術者資格制度(JSTQB)技術委員、ソフトウェア・シンポジウム(SS)プログラム委員
2013～ソフトウェア品質(SQiP:Software Quality Profession)委員会運営委員 など

提案の背景



派生開発分野ではPFDによるプロセス記述とそれに基づくプロセス設計運営が提案されていますが、その実現は容易ではありません。

プロセス設計はさらにその先の話...

例：現状のプロセスを表現する、に関して

- 自らのプロセスを表現することができない組織やチームが意外と多い
実は自分達の現状を把握していない／書き方が分からない組織も多い
- 仮にプロセスらしきものを表現できたとしても

① 現状を表していない

② 現状を表しているが抽象度が高すぎ・低すぎ・バラバラ混在

例：「設計→実装→テスト」のように一般論化する／書いてあっても活用できない

③ 現状を的確に表現できても、それだけではどこが問題点なのかが分からない

・これまでの慣習＝明確な意図を持たない作業方法であるため

・.....などなど表現しても結局活用しない／活用できない(放置される)

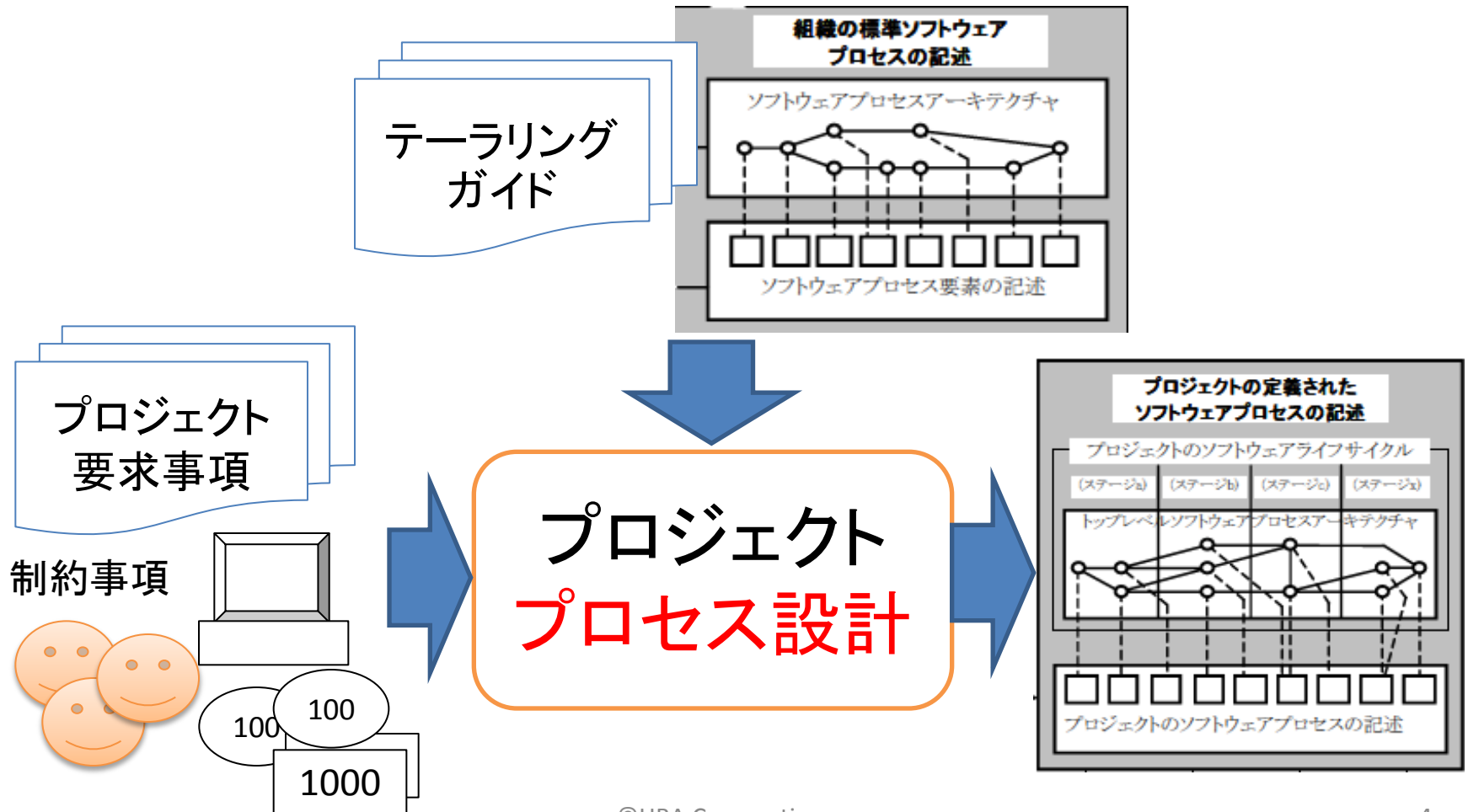
一般論をたくさん並べて無駄に重たい内容になることも

多大な工数と期間をかけて作成したのに活用されないことも多い

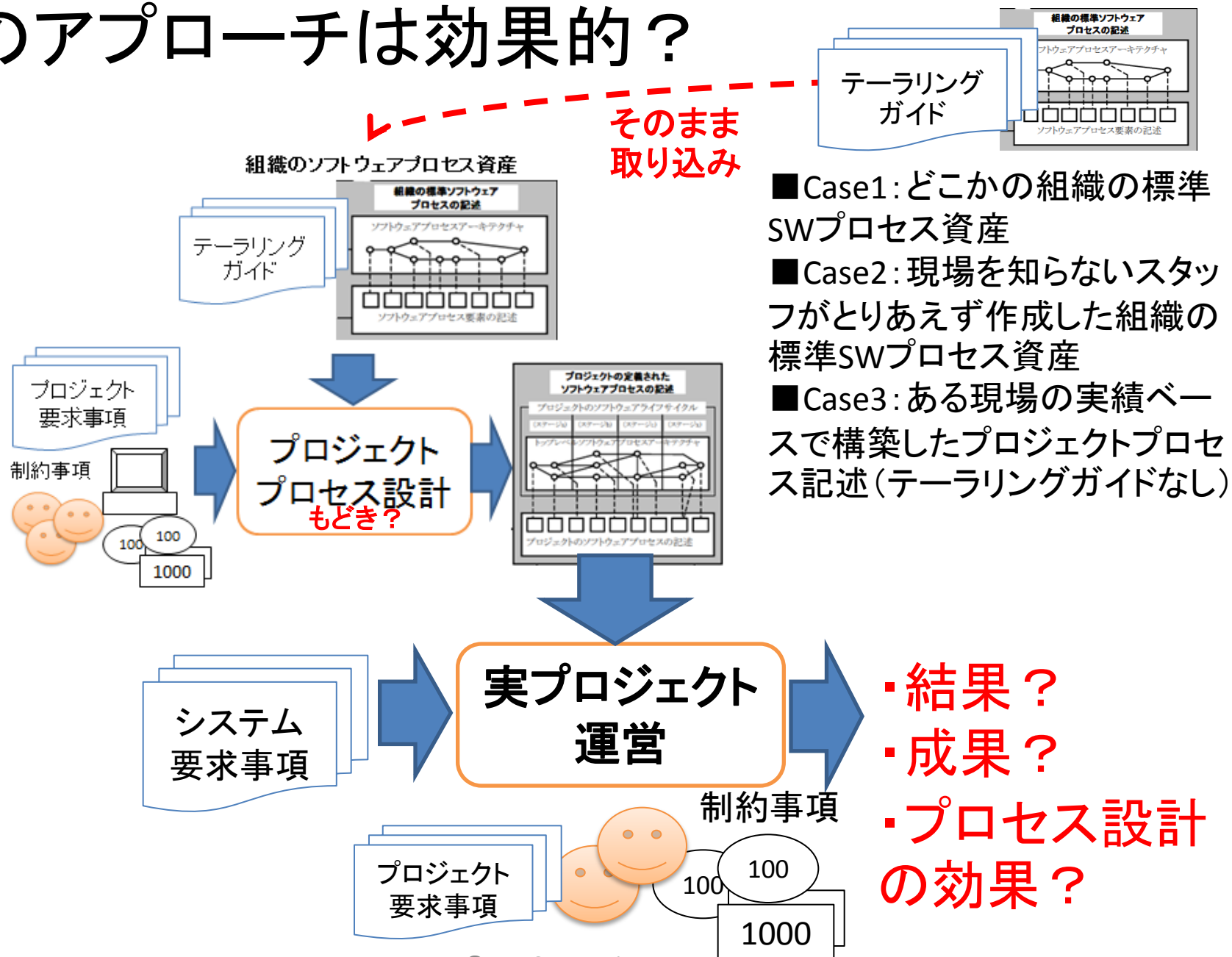
組織能力成熟度Level3以上ではこんなことができる(らしい)

能力成熟度モデルのキープラクティス 1.1版
(CMU/SEI-93-TR-25 公式日本語版)より

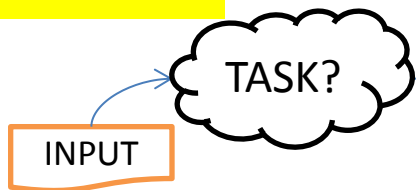
組織のソフトウェアプロセス資産



このアプローチは効果的？



BEFORE



? 改善の鉄則？

AFTER



- 製造業においては、改善効果を高めるために“現状のプロセス・改善後のプロセスを表現すること”が常識となっている。（と聞いている）
- 品質システム（当時）導入時の外部コンサルタントも同じ対応をわれわれに求めてきた。
 - ①現状のプロセスをフローチャートに明記
 - ②要求事項へ適合するようにプロセスを直す（？）

**事例：①で現状ではないプロセスを記述して失敗
→その後5年間その解消を含めて迷走**

プロセス表現を誤ると



誤プロセス記述を実際のプロセスに書き直すために、そして問題点を洗い出すために別途ヒアリングや現状調査などの多くの時間と手間が必要になる。

→現状を表す／表さないにかかわらず完成した「プロセス表現」の活用方法が分からないため、実際には普段の業務運営上で活用されないまま放置される、または& 形式的に活用したことにする対応が多くなる。

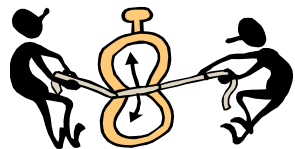


→手間がかかる割に効果がない＝無駄だ、という認識を現場に植え付ける

→以降の業務運営や改善がうまく進まない原因になってしまう。



工数・時間をかけたのに放置される、誤認識を与えるなどいろいろよくない影響が・・・ではどうすればいいの？



＜提案＞

問題構造分析とPFDの併用 その実現への段取り(事例)

解決に必要なアプローチ

一般解

理解

今回の提案

現状のプロセス(と成果)を
的確に表現できるようになる

そして

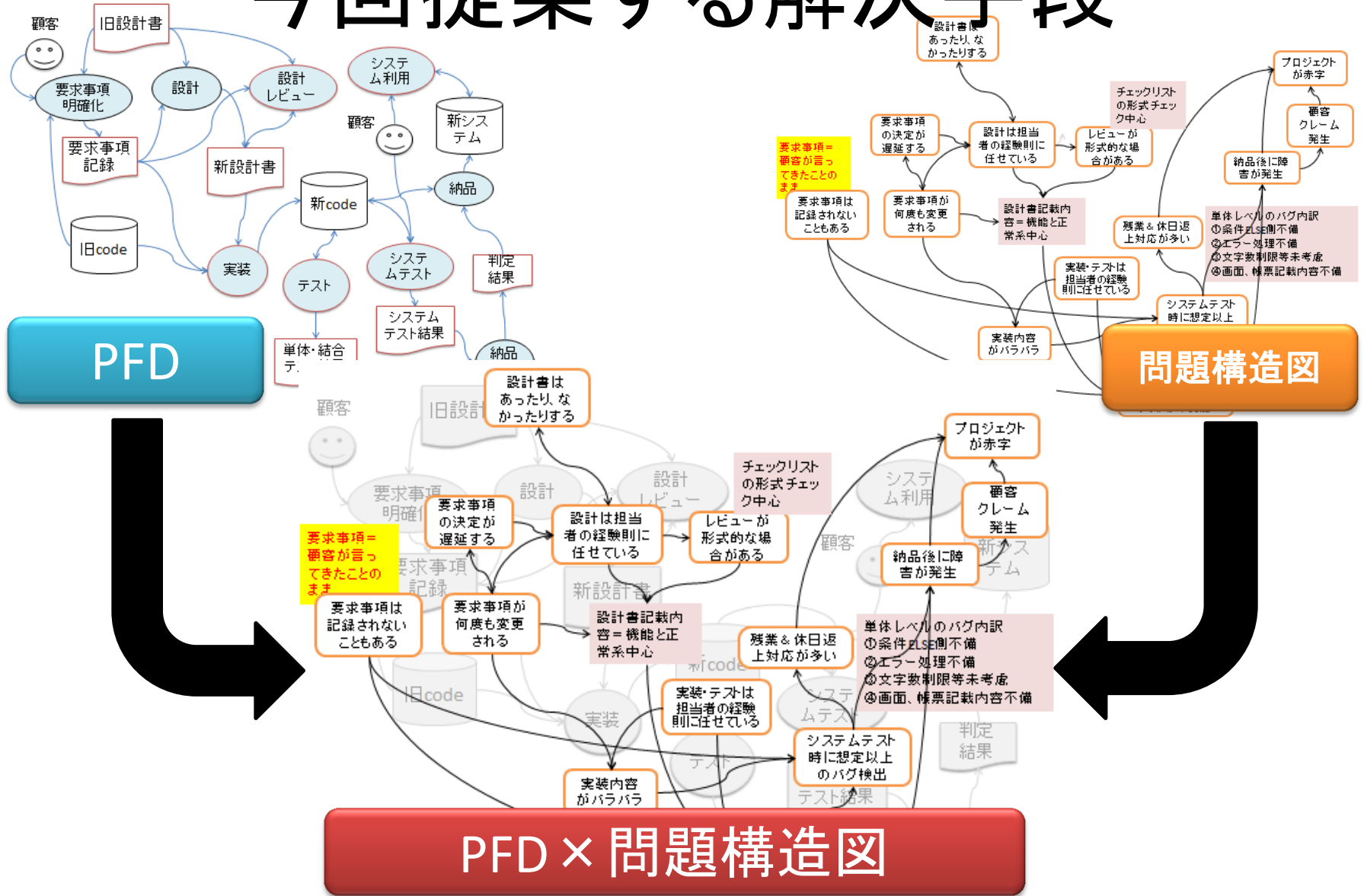
応用

外部要求や制約事項に応じて
プロセスを**自在にデザイン**する

①改善
実践実
績から
仮表現

②相互連携・
実践で表現
を洗練して活
用・効果を高
める

今回提案する解決手段

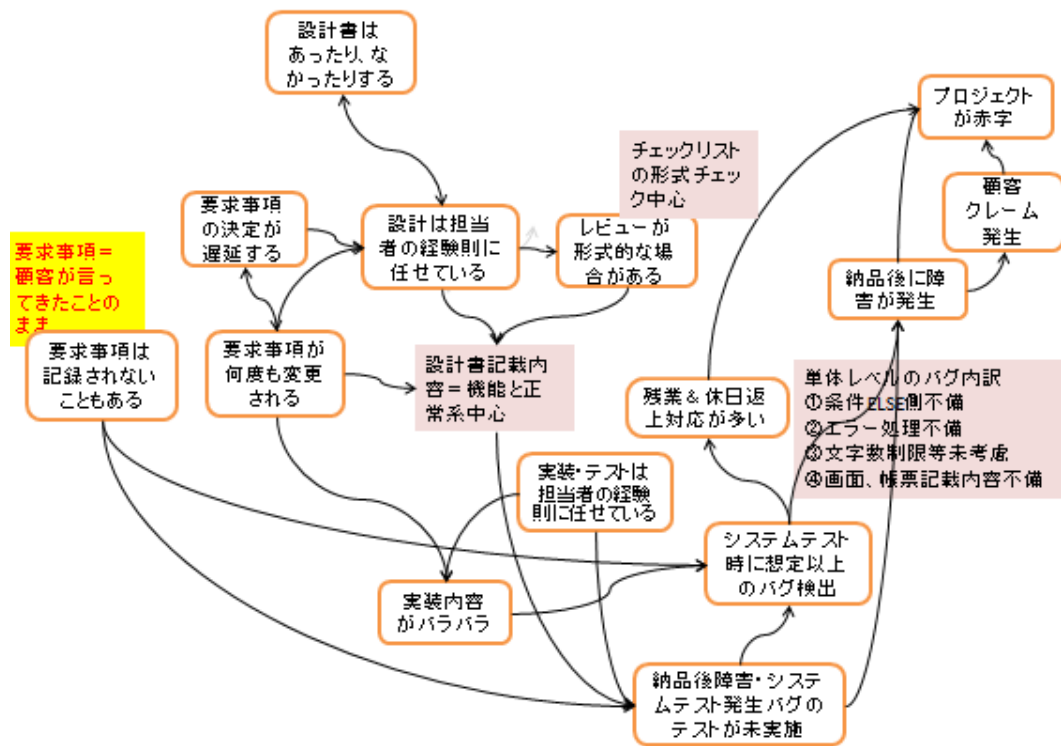
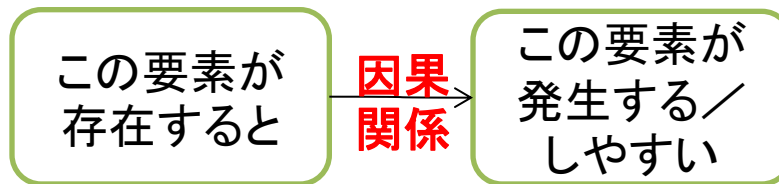


問題構造図

システムズアプローチ

関係者・現場が持つ課題・問題事項(事実情報)をベースに、因果関係による問題発生構造(問題発生システムの構造)を明確化した図

関係者が参画してこの図とコンテキストを共有し、解決を図るシステムズアプローチで使われる



自ら提示し明確化した問題構造から自分達で決めた解決手段を実践するため
自律した改善運営を促進しやすい特徴がある

＜SaPID・SPINA3CH自律改善メソッド等で採用されている＞

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

SPINA3CH: http://sec.ipa.go.jp/reports/20130326_3.html

改善サイクル①(1):

リーダ・実務担当者に**各自が感じている問題点**を出してもらう
→事実で裏を取り具体化・一般論や噂、思い込み、決めつけは排除

具体化事例

プロジェクトの
収支が赤字

要求事項の
決定が遅延
する

毎日残業&休
日返上対応が
多い

テストは担当
者の経験則に
任せている

要求事項が
何度も変更
される

詳細設計書
記載レベル
がバラバラ

システムテス
ト時に大量
バグ検出

納品後に多くの
障害が発生(単
体テストレベルの
ものが多い)

設計時変更85件
実装時変更53件

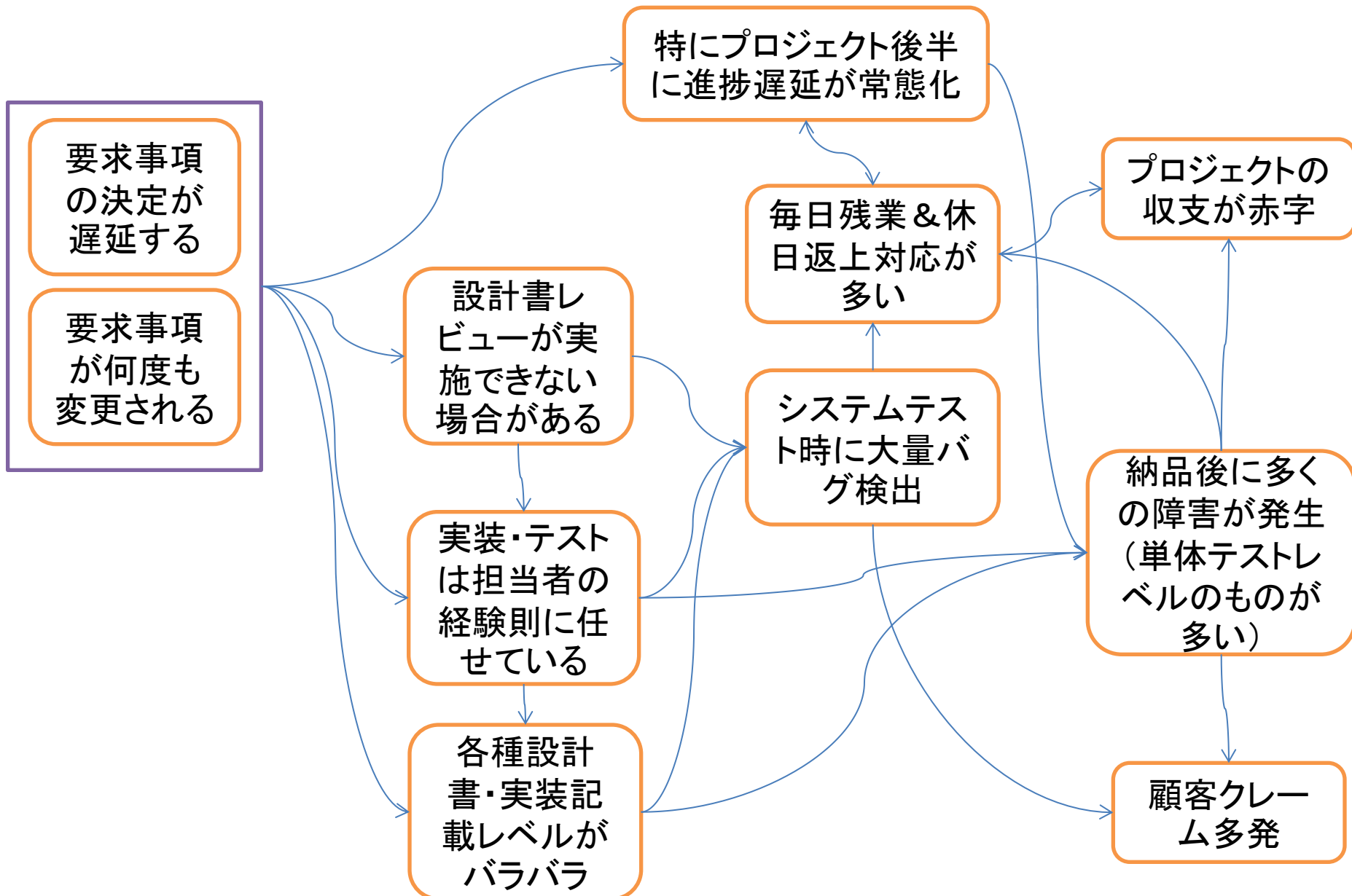
プロジェクト
後半-進捗遅
延が常態化

レビューが実
施できない
場合がある

詳細設計レビュー
実施74/92

顧客クレーム
多発

改善サイクル①(2): 問題事項の因果関係を粗く整理



改善サイクル①(3):

詳細確認 → **改善成果領域と改善対象を特定** → **改善実践**

要求事項記載内容
=顧客が言ってき
たことそのまま

要求事項
の決定が
遅延する

要求事項
が何度も
変更される

要求事項は
記録されな
いことが多い

担当者が用

この時点ではPFDやフロー
チャートなどのプロセス表現
を無理に作らないなど、改善
運営の軽量化を重視する。

設計書は
あったり、な
かったりする

設計書記載内容 =
機能と正常系中心

設計書レ
ビューは時間
があればやる

実装・テスト
は担当者の
経験則に任
せている

実装内容
バラバラ

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

プロジェクトの
収支が赤字

システムテス
ト時に大量バ
グ検出

納品後に
多くの障
害が発生

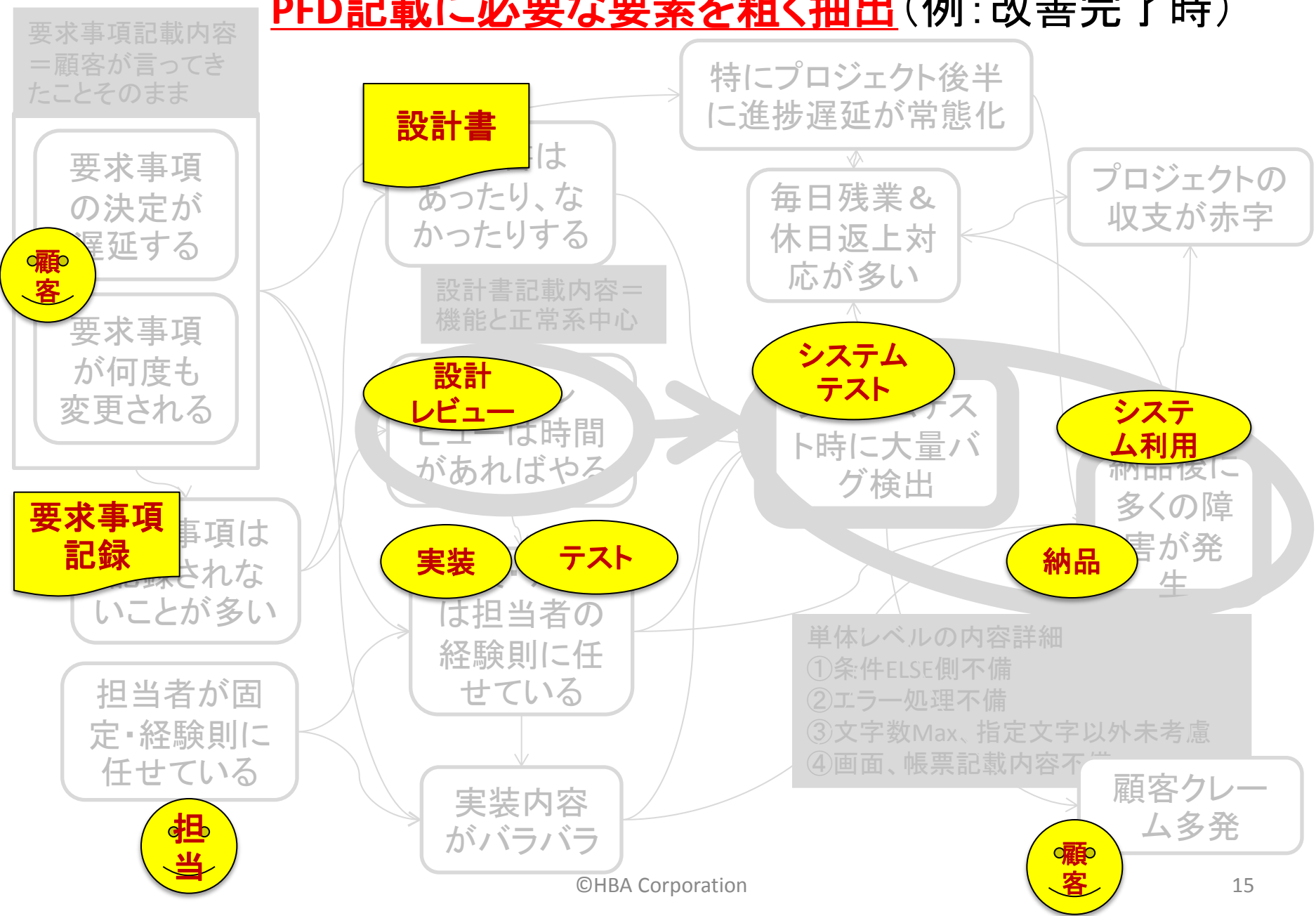
単体レベルの内容詳細

- ①条件ELSE側不備
- ②エラー処理不備
- ③文字数Max、指定文字以外未考慮
- ④画面、帳票記載内容不備

顧客クレーム多発

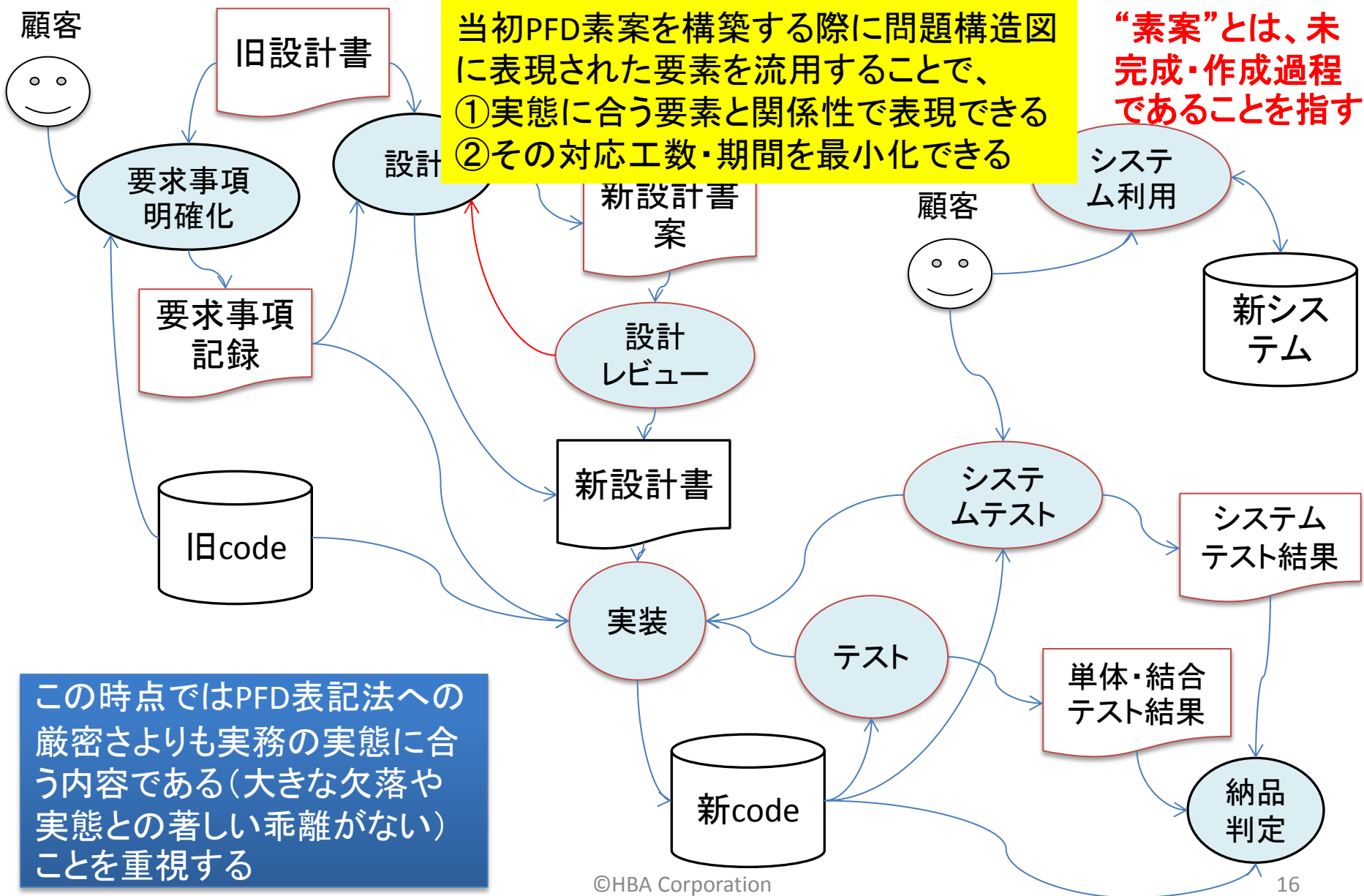
改善サイクル①(4):

PFD記載に必要な要素を粗く抽出 (例:改善完了時)



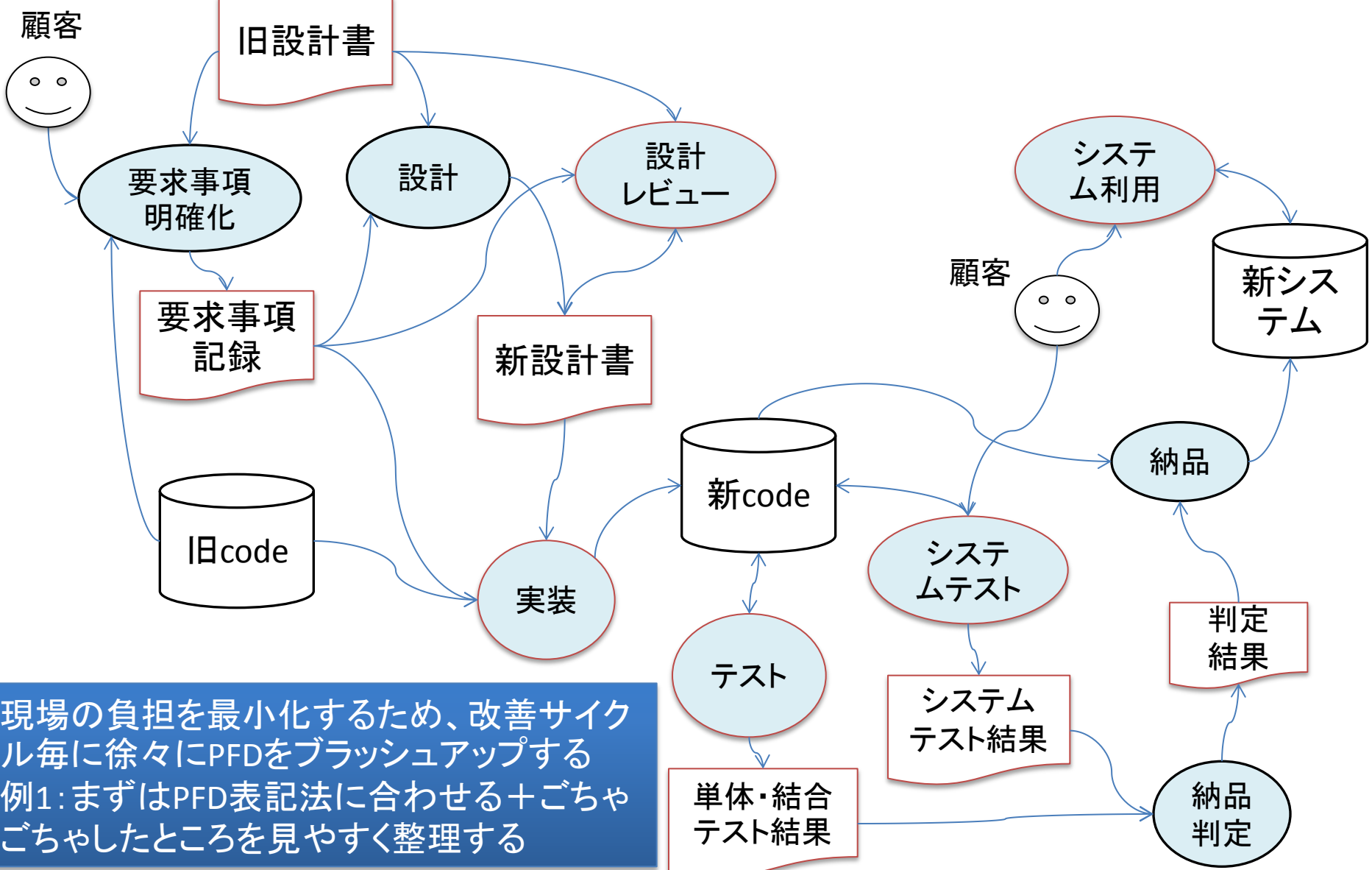
改善サイクル①(5):

抽出したPFD要素とその周辺の確認結果からPFD素案を作成



改善サイクル②:

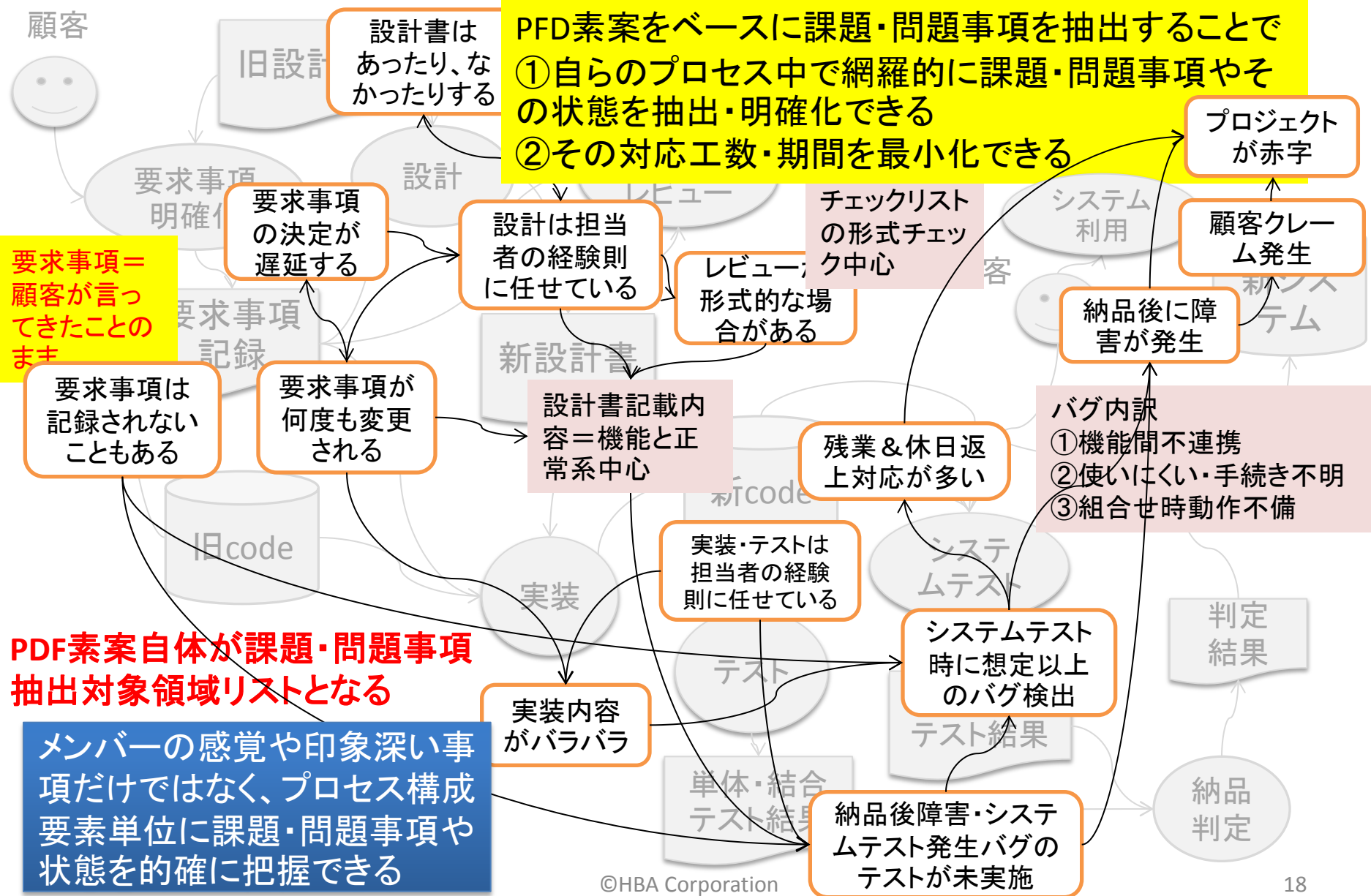
改善サイクルを回しながら PFD素案をブラッシュアップ(1)



現場の負担を最小化するため、改善サイクル毎に徐々にPFDをブラッシュアップする
例1: まずはPFD表記法に合わせる+ごちゃごちゃしたところを見やすく整理する

改善サイクル③以降(1):

ブラッシュアップしたPFDを併用して課題・問題事項を抽出し分析



顧客
旧設計
設計書はあつたり、なかったりする

PFD素案をベースに課題・問題事項を抽出することで
①自らのプロセス中で網羅的に課題・問題事項やその状態を抽出・明確化できる
②その対応工数・期間を最小化できる

要求事項 = 顧客が言ったことのまま

要求事項の決定が遅延する

設計は担当者の経験則に任せている

レビュー: 形式的な場合がある

チェックリストの形式チェック中心

プロジェクトが赤字

顧客クレーム発生

要求事項は記録されないこともある

要求事項が何度も変更される

設計書記載内容 = 機能と正常系中心

残業 & 休日返上対応が多い

バグ内訳
①機能間不連携
②使いにくい・手続き不明
③組合せ時動作不備

PDF素案自体が課題・問題事項抽出対象領域リストとなる

メンバーの感覚や印象深い事項だけではなく、プロセス構成要素単位に課題・問題事項や状態を的確に把握できる

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想定以上のバグ検出

実装内容がバラバラ

納品後障害・システムテスト発生バグのテストが未実施

判定結果

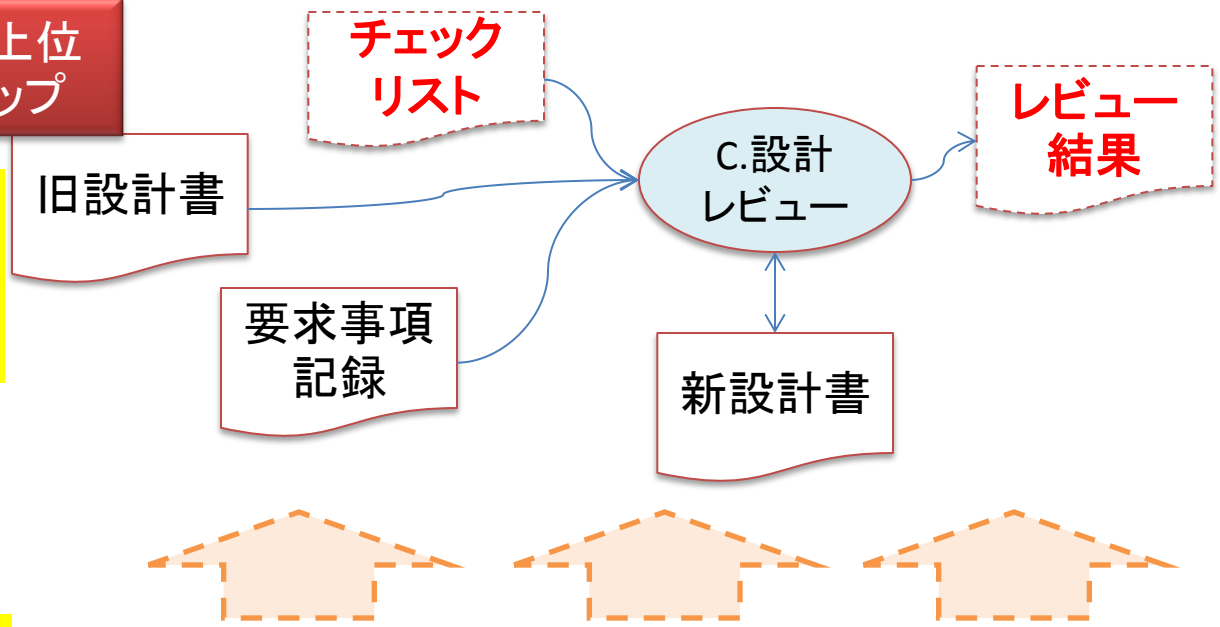
納品判定

改善サイクル③以降(3) :

改善サイクルを回しながら PFD素案をブラッシュアップ(2)

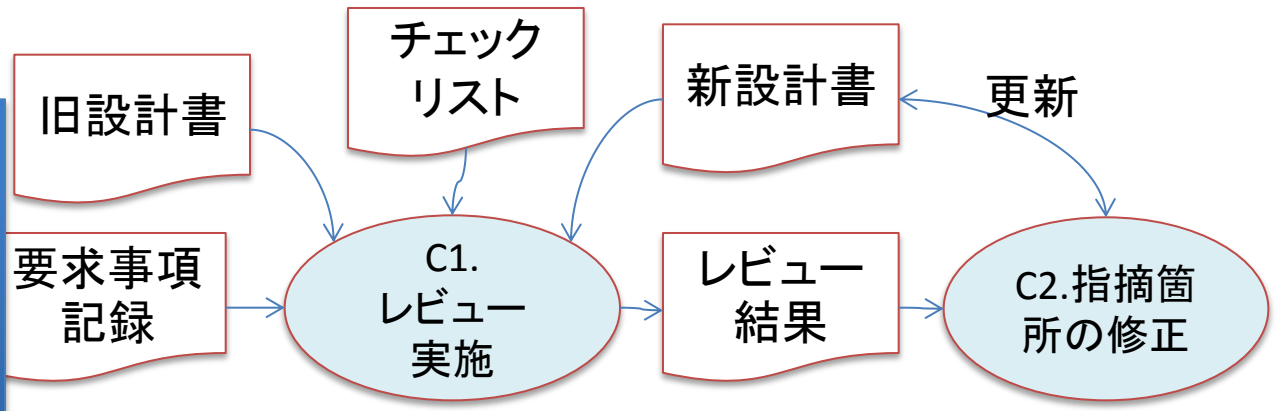
【事例】改善対象深掘～上位レイヤ表現ブラッシュアップ

【上位レイヤ表現】
②整合確保対応
⇒赤文字部分



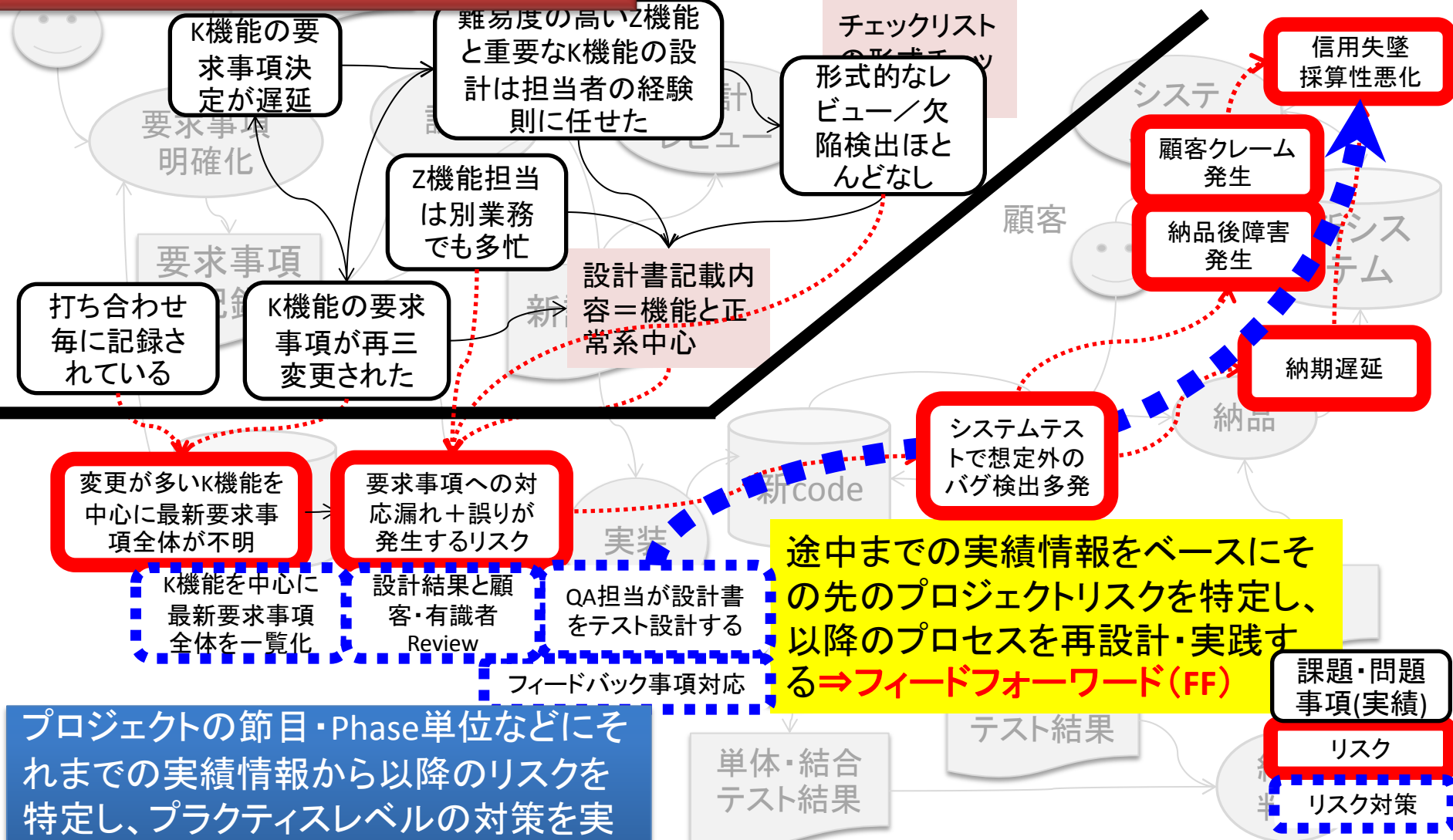
【下位レイヤ表現】
①改善対象の詳細化

現場の負担を最小化するため、改善サイクル毎に徐々にPFDをブラッシュアップする
例2:改善対象領域は下層レイヤまで明らかにし、上下レイヤ間の整合を確保する



改善サイクルn(1): 進捗管理・ふりかえり活用→次フェーズ以降プロセス再設計して実践

【事例】設計Phase完了ふりかえり時



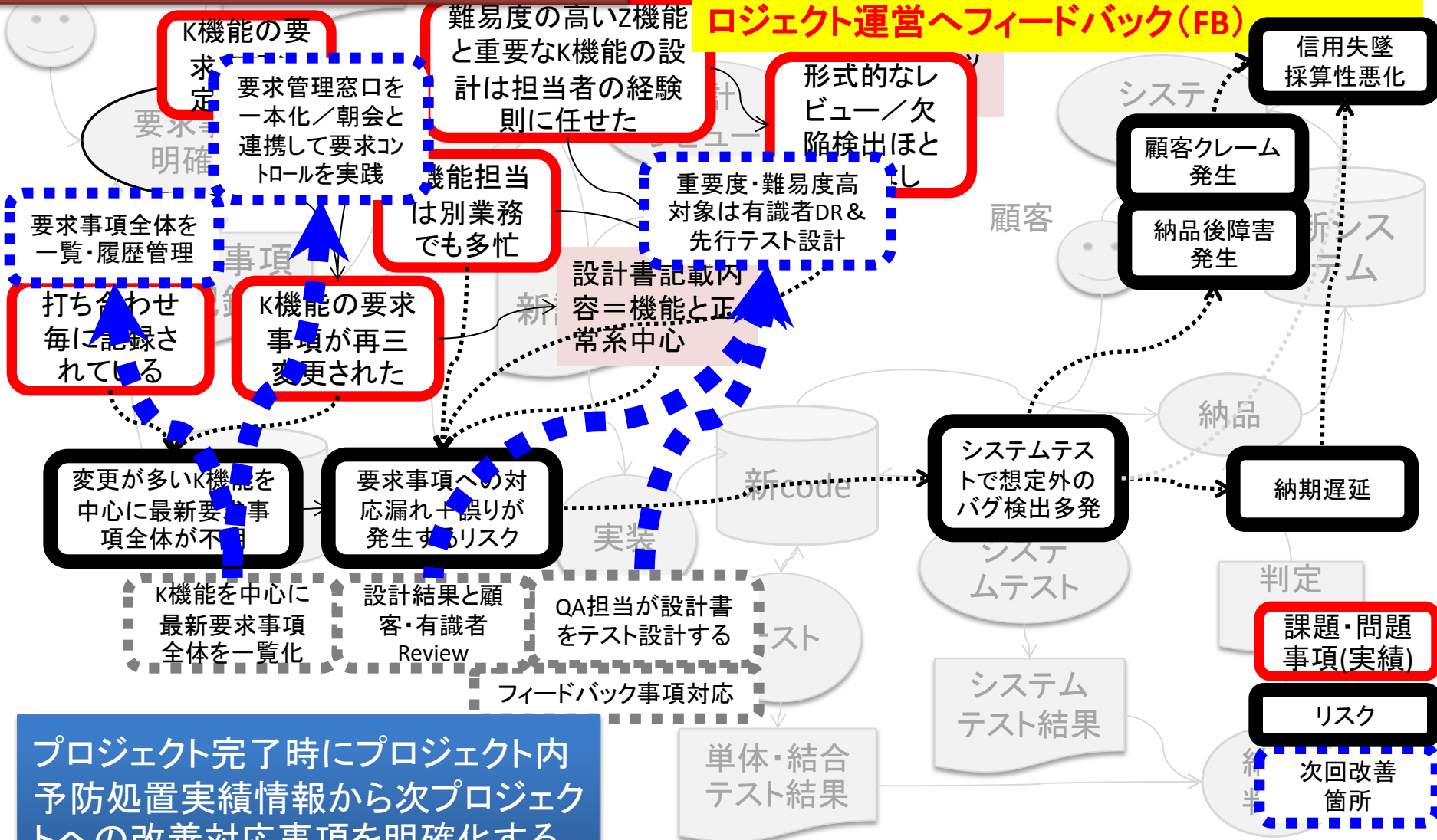
プロジェクトの節目・Phase単位などにそれまでの実績情報から以降のリスクを特定し、プラクティスレベルの対策を実践する(プロジェクト内予防処置)

改善サイクルn(2):

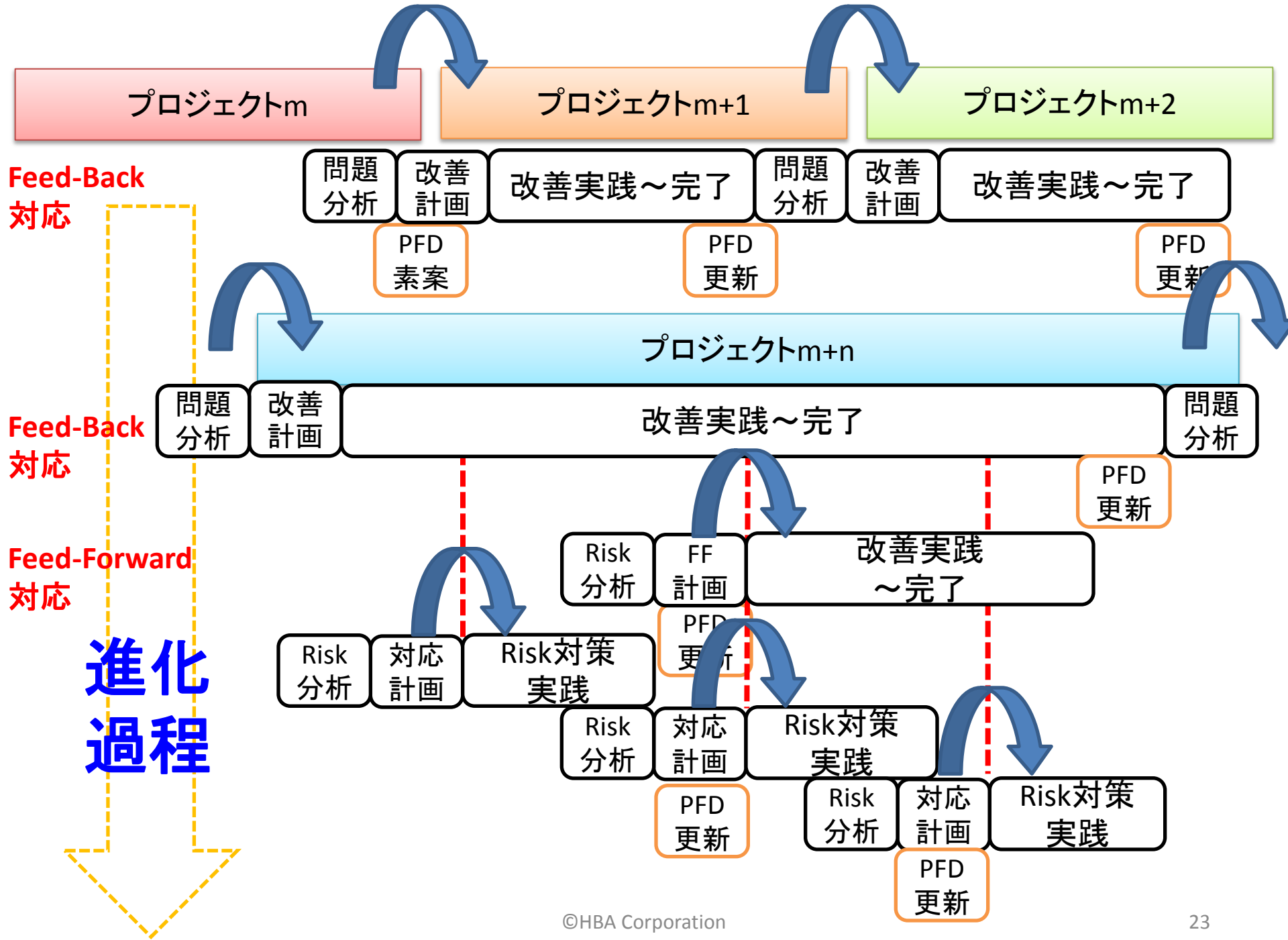
Project完了ふりかえり活用→次プロジェクトへ改善対応事項明確化

【事例】Project完了ふりかえり時

フィードフォワード(FE)情報を利用して次プロジェクト運営へフィードバック(FB)



プロジェクト完了時にプロジェクト内予防処置実績情報から次プロジェクトへの改善対応事項を明確化する



プロセス設計実践に必要なこと

なぜ問題構造分析がプロセス設計に役立つのか？

プロジェクトプロセス設計

プロジェクトプロセス設計とは、プロセス・タスクの意味・意義を把握・理解した上で、プロジェクト／製品・サービス要求事項や制約条件を考慮して

①要求事項を満足する製品／サービスをどのように構築・提供するのか

②プロジェクト運営上で発生する可能性のある負の事象を先読み(未来を予測)し、そのリスクレベルに見合う現実的&有効な対策をどの段階で実施するのかを明確化し、プロジェクト計画として統合すること。

今回は特に下線部に必要な事項を明確化

プロセス・タスクの意味・意義

プロセス・タスクの意味・意義は、
単一プロセス・タスクだけでは把握できない



そのプロセス・タスクがどのプロセス・タスク
にどのような影響を与えるのか
=相互関係性が分からないと把握できない

プロジェクト監視・
制御プロセス

監視

制御

評価

終了

要求定義プロセス

要件の定義

要件の評価・レビュー

結合プロセス

結合

テスト準備・結合の評価

方式設計プロセス

方式の確立・文書化

方式の評価・レビュー

複数の(プロセス・タスク)間の
相互関係性(影響)把握のため
に因果関係の理解が必要
→これがないとPFD表現がで
きて使いこなせない

事例

プロジェクトの開始時に「このあと大風が吹く」という情報が入手できたとします

「風が吹く」からの構造的先読み

風が吹く

土埃が舞う

的確なプロセス設計実践には、まだ発生していない未来の出来事を想定することが不可欠

構造的先読みのために鍵となる要素



土が目に入る

盲人が増える

三味線が売れる

三味線屋が儲かる

盲人の職業は三味線弾きが多い

盲人は三味線を買う

三味線には猫皮必要

ダイナミックにプロセス設計ができる技術者・管理者は、プロジェクトコンテキスト・製品特性を元に「風が吹く」情報により図のような問題・リスク構造を想定し、その対策を含めたプロセスを的確に設計できる
⇒そうなるためには？

猫が減る

先読みはフィードフォワード(FF)技術

ネズミが増える

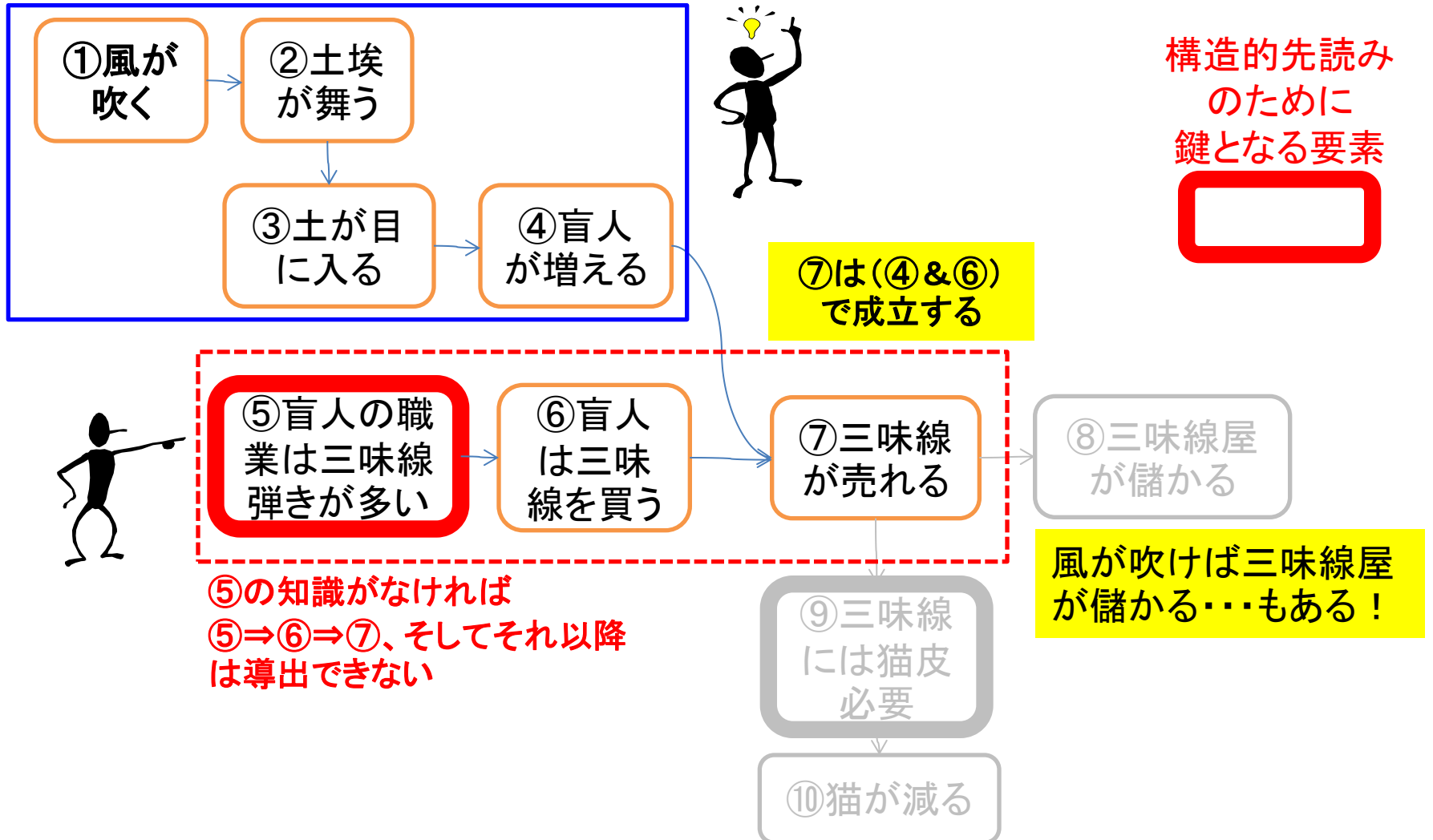
ネズミが桶をかじる

桶の需要が増える

桶屋が儲かる

リスク・問題の構造的先読み(1)

①⇒②⇒③⇒④は
比較的容易に想定できる(③⇒④はちょっとムリがある?)

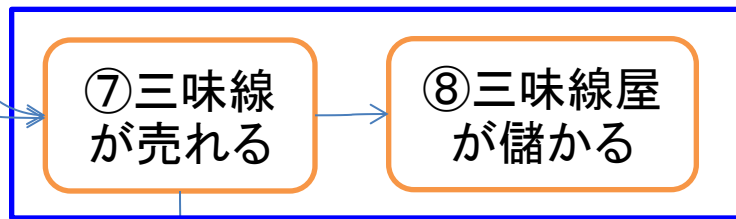


リスク・問題の構造的先読み(2)

①～④+⑤～⑦がクリア
されたと仮定します

構造的先読みのため
に鍵となる要素

⑦⇒⑧は容易に想定できる



以上のように、リスク・問題を誘発する
情報(ここでは「風が吹く」)からリス
ク・問題発生構造を組み立てる



鍵となる要素の知識がなければ構造的
に全体を先読みできない
⇒事前対策(プロセス設計)できない
⇒事後対応するしかなくなる

⑨ 三味線
には猫皮
必要

⑨の知識がなければ
⑨⇒⑩⇒⑪は導出できない

⑩ 猫が減る

⑪ ネズミ
が増える

⑫の知識がなければ⑫⇒⑬⇒⑭は導出できない

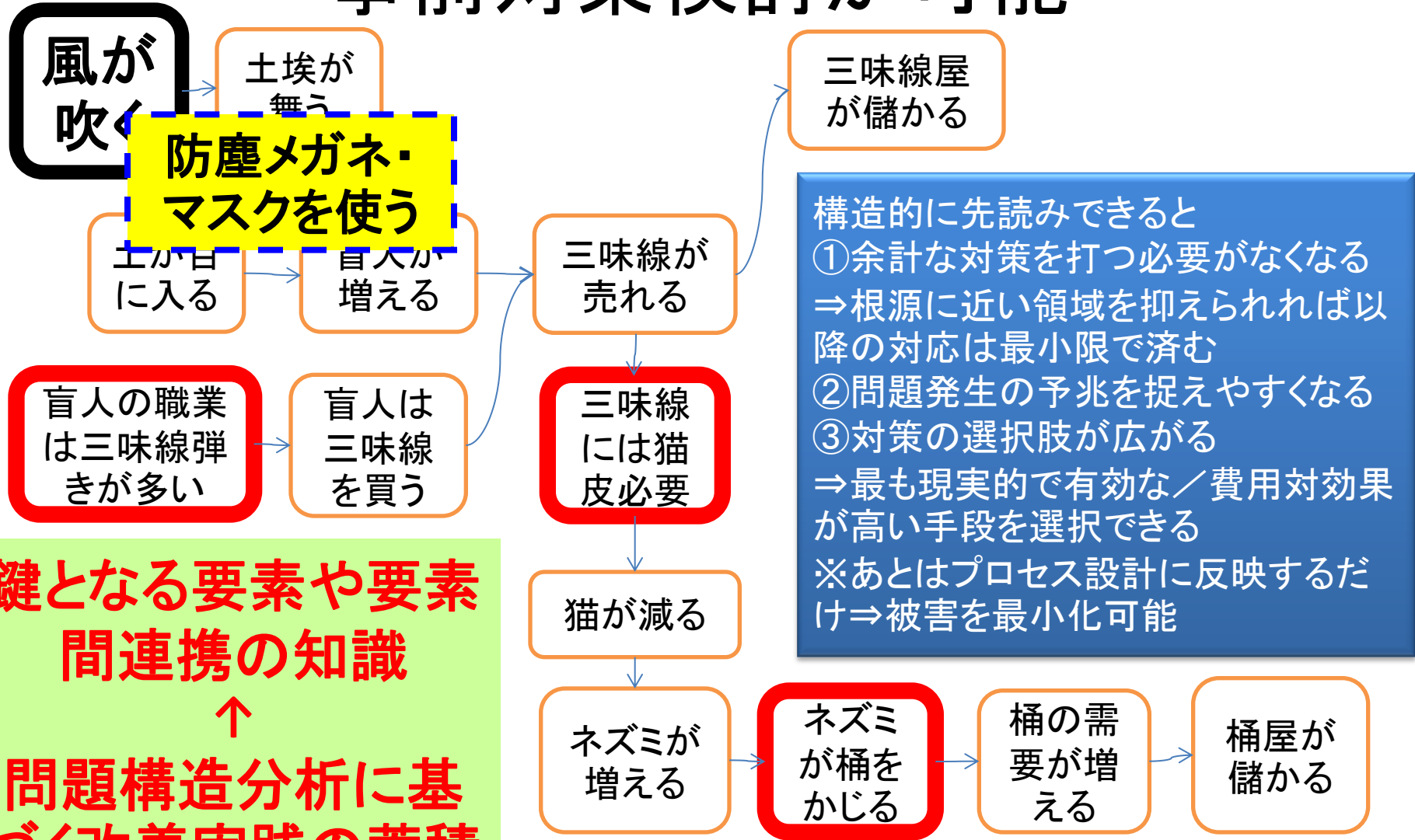
⑫ ネズミ
が桶をか
じる

⑬ 桶の需要
が増える

⑭ 桶屋が
儲かる

プロジェクト運営の中でリスク・問題構造明確化→改善
対応やリスク対策を継続実践することで鍵となる要素を
含めた知識・ノウハウを獲得・蓄積できる

先読みできれば効果的な 事前対策検討が可能



鍵となる要素や要素
間連携の知識

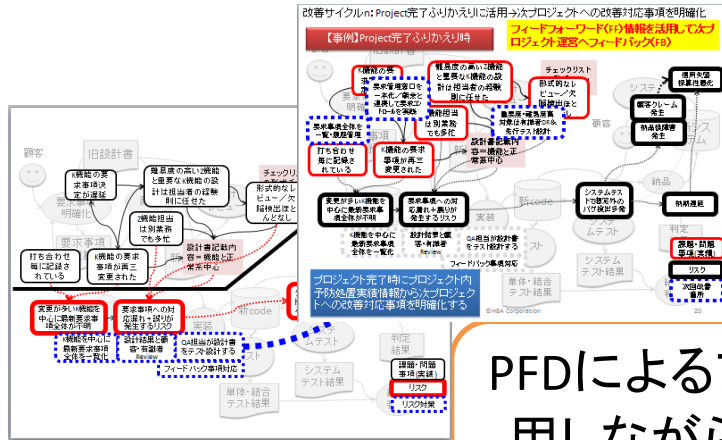


問題構造分析に基づき
改善実践の蓄積

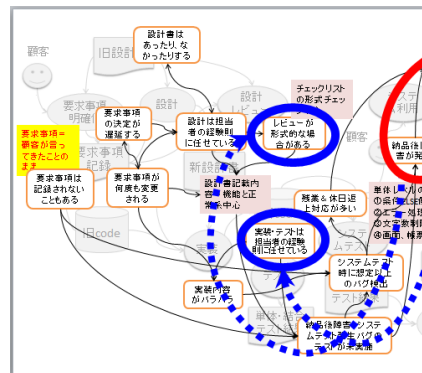
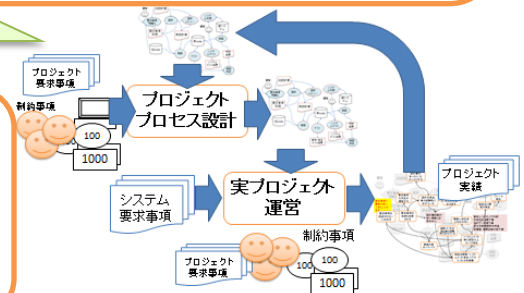
まとめ

当アプローチをまとめると

改善を継続→ノウハウが蓄積されることで先読み力が強化され、PFDによるプロセス設計が実践でき、効果を獲得できる



PFDによるプロセス表現を活用しながらより効果的な改善を継続的に実践できる

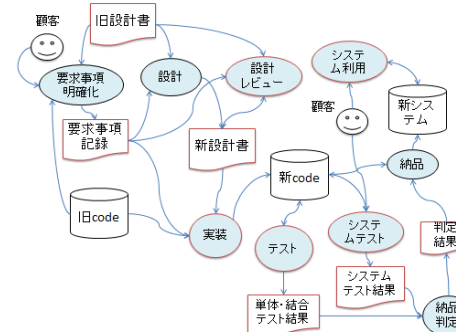


改善を回しながらPFDによるプロセス表現をよりの確な内容にすることができる



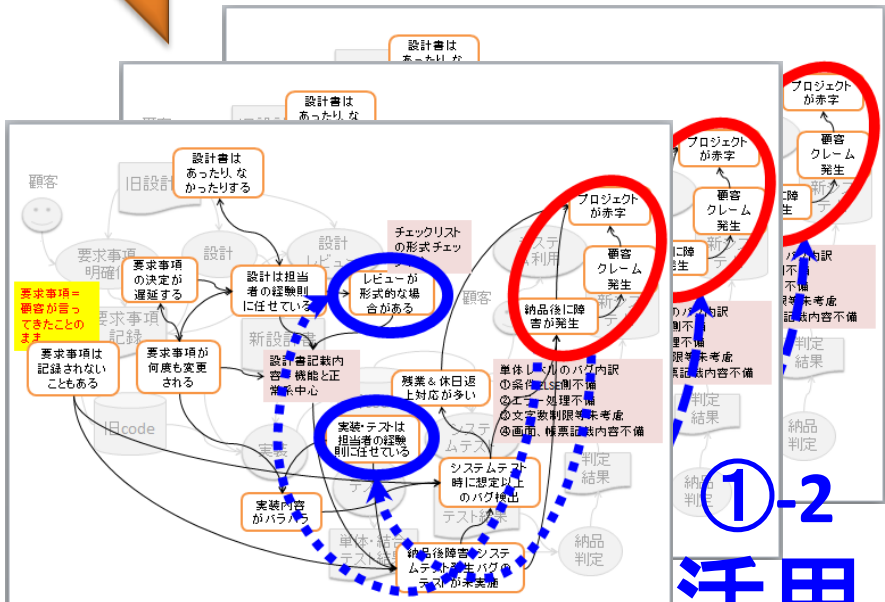
課題・問題事項をベースにした問題構造図よりPFD要素を抽出して表現する

最小の工数でプロジェクトの実態をPFDで表現できる



FB→FFの遷移／併用による 先読み力の強化

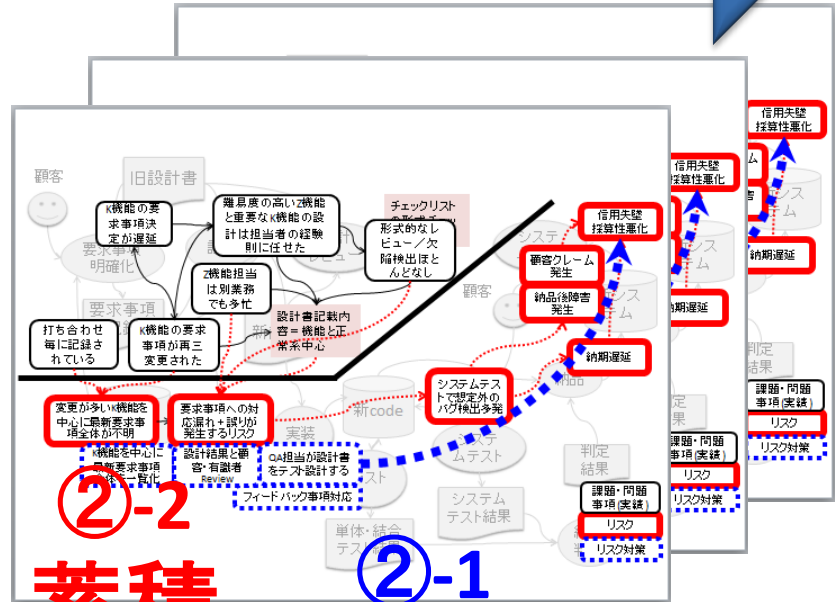
① Feedback



①-1 蓄積

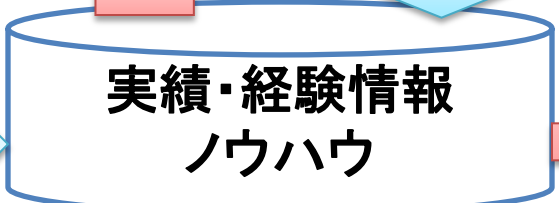
①-2 活用

② Feedforward



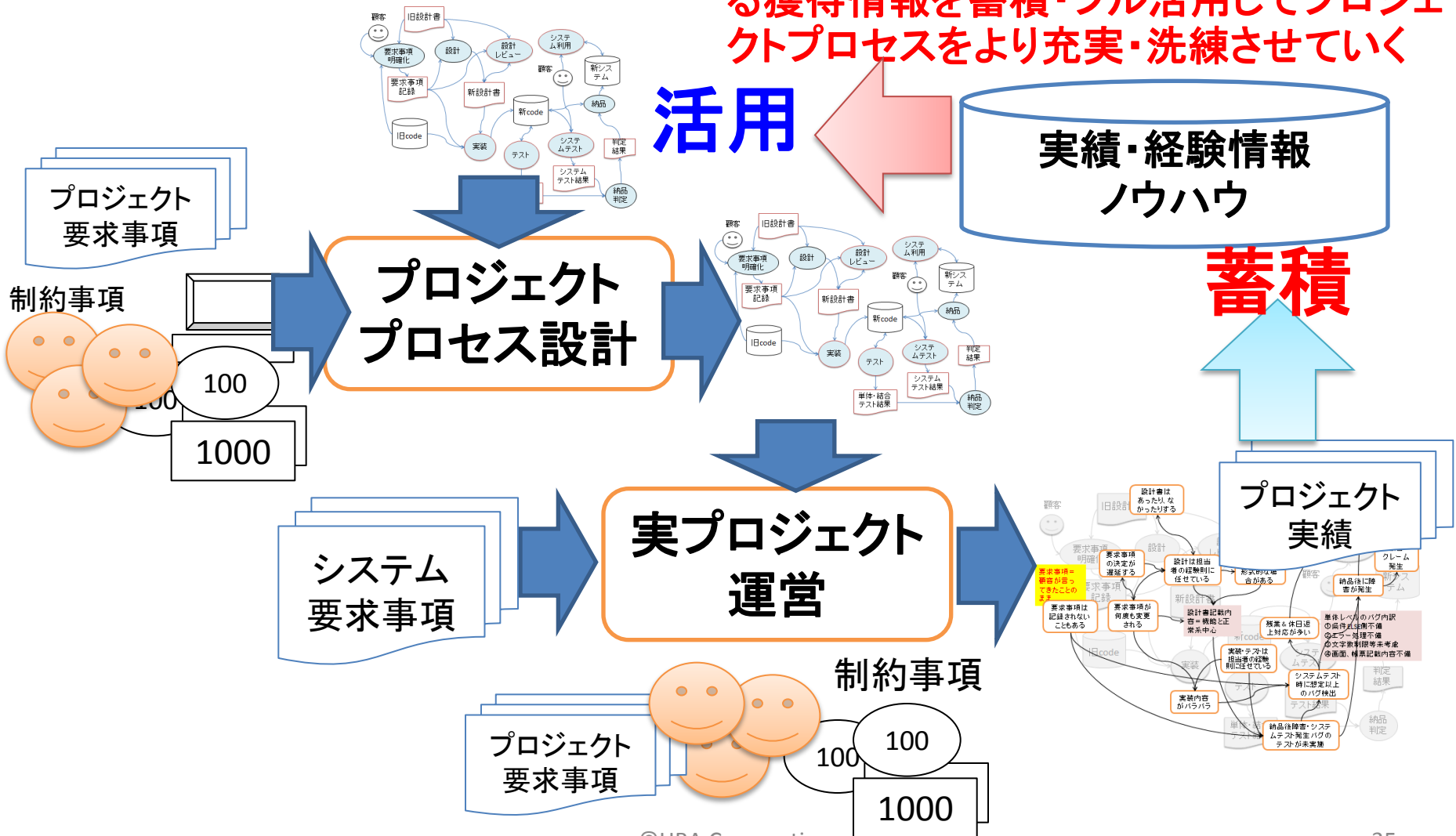
②-2 蓄積

②-1 活用



実績をベースに実効性／競争力のある プロセス設計実践へ

自らの計画⇒運営⇒経験⇒改善実践による
獲得情報を蓄積・フル活用してプロ
ジェクトプロセスをより充実・洗練させていく



当手法の活用方法と今後の課題

- 当手法の活用方法:

実際のプロジェクト・業務運営実践に
個人・チーム・組織の改善継続運営に
改善推進役やSEPG/PMOの実務とトレーニングに
Project Manager等管理者の実務とトレーニングに

- 今後の課題:

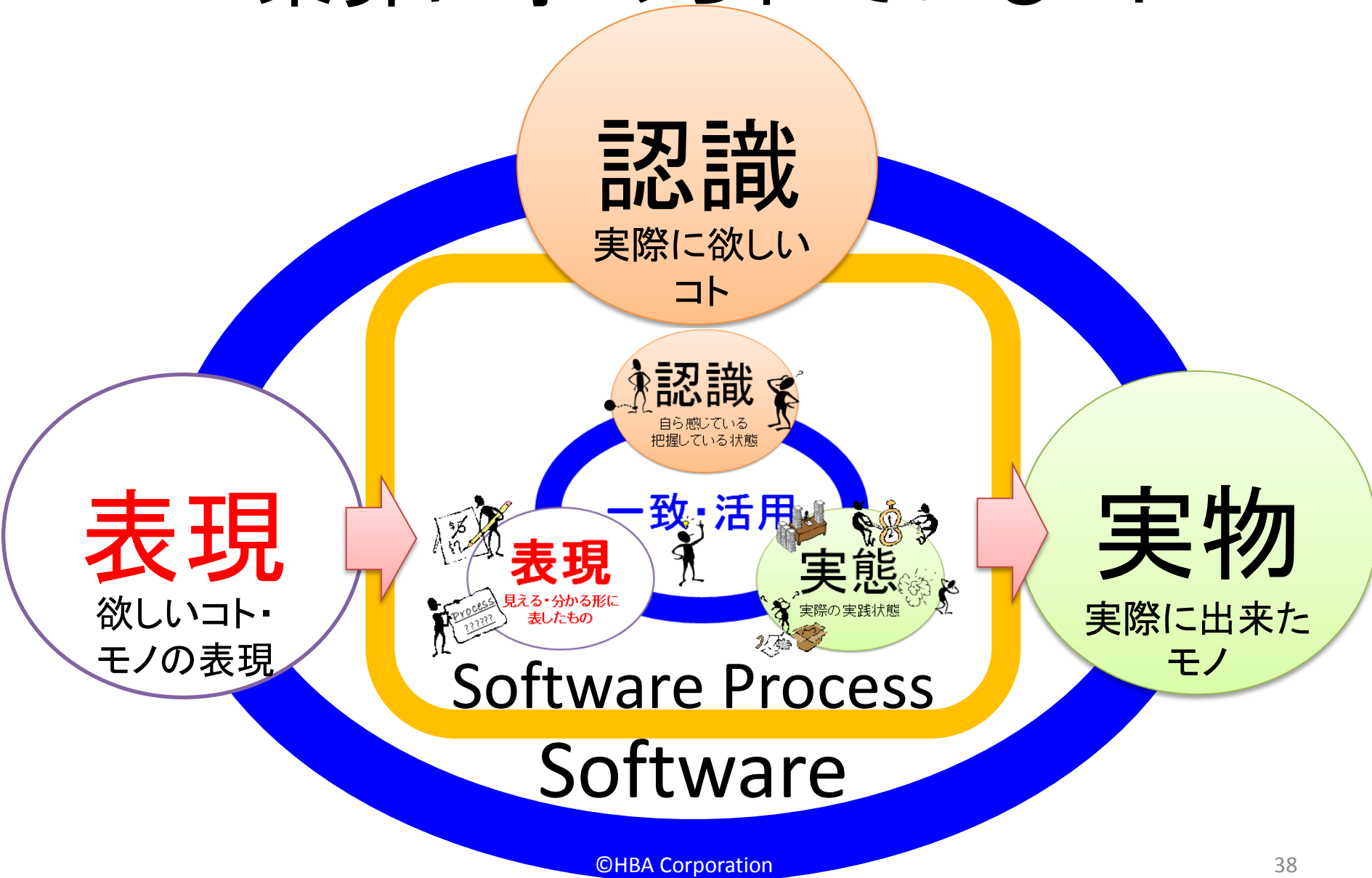
組織レベルの効果的なソフトウェアプロセス
管理(Level3以上)は別考察・別整理が必要

Software Processに求められること

知行合一 ⇒ 知**表**行合一



IT業界に求められているコト



参考文献

- PFD(Process Flow Diagram)の書き方 第3版 2009/9/21
http://homepage3.nifty.com/koha_hp/process/PFDform3.pdf
- 能力成熟度モデルのキープラクティス 1.1版
(CMU/SEI-93-TR-25 公式日本語版)
<http://www.sea.jp/CMM/publish/CMM-J99.html>
- 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 (Systems analysis / Systems approach based Process Improvement method) 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
- 2013.3.26 当事者の問題意識に基づくプロセス改善手法
「SPINA³CH自律改善メソッド」の改訂
http://sec.ipa.go.jp/reports/20130326_3.html
- 共通フレーム2013～経営者、業務部門とともに取組む使えるシステムの実現
独立行政法人情報処理推進機構 (IPA) 技術本部
ソフトウェア・エンジニアリング・センター (SEC)

ご清聴ありがとうございました

この提案内容を業界内で有効活用できればうれしいです

当事例の問い合わせ先はこちらです

株式会社HBA 安達賢二 : adachi@hba.co.jp ・ 猪股宏史 : inomata@hba.co.jp