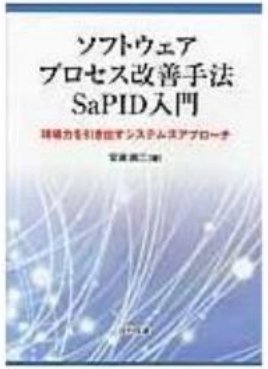
Full Version

システム思考：構造化・見える化による課題発見解決
～SaPIDを活用した研究へのアプローチ



書籍の該当頁数

P.00

ソフトウェア品質管理研究会 研究コース2「ソフトウェアレビュー」アドバイザー
Software Quasol
安達 賢二
https://www.softwarequasol.com/
adachi@hba.co.jp

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

安達 賢二（あだち けんじ） adachi@hba.co.jp
株式会社HBA 経営企画本部 Exective Expert（理事）
イノベーション推進室 アドバイザー
http://www.softwarequasol.com/
株式会社Levii 共創ファシリテーター
https://levii.co.jp/about/

きたのしろくま
@kitanosirokuma
Twitter（X）

【経歴】
2012年社内イントレプレナー第一号事業者として品質向上支援事業を立ち上げ。
自律運営チーム構築・変革メソッドSaPIDをベースに、
関係者と一緒に価値あるコトを創る共創ファシリテーター／
自律組織・人財育成コーチとして活動中。

【社外活動】
NPO法人 ソフトウェアテスト技術振興協会（ASTER）理事
JSTQB（テスト技術者資格認定）技術委員
JaSST（ソフトウェアテストシンポジウム）北海道
　2006-2009実行委員長　2010-2018実行委員　2019～2022サポーター
JaSST-Review（ソフトウェアレビューシンポジウム）実行委員
JaSST-nanoお世話係
ASTER研究事業「レビュー体系化」メンバー
テスト設計コンテスト本部審査委員(2015-2017)
SEA（ソフトウェア技術者協会）幹事・北海道支部メンバー
SS（ソフトウェア・シンポジウム）プログラム委員
第33-39期SQiP研究会レビュー分科会アドバイザー
SQuBOK_Ver3プロセス改善研究Grリーダ（with プロセス改善の黒歴史研究）
TEF北海道お世話係／TOCfE北海道幽霊メンバー　など

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

今日の講義の概要

- SQiP研究会などで研究を進める際に、**最も苦労するのが「研究する対象を決める」こと**です。
- これは普段の実務に存在する様々な問題点の中から、取り組むべきもの／取り組む価値のあるもの＝課題を探す、研究目的を特定する作業と言い換えることができると思います。
- 取り組むべき課題が見つかれば、解決手段を考えるハードルはそう高くはないことも多いです。
- 昨今のDX化を目指す取り組みにおいても「そもそも何をしたいのか？」が出せずに右往左往するケースがありますがこれも同じく課題が特定できないケースです。
- また、IT開発プロジェクトで顧客がそもそも何をしたいのかがわかっていないため、要件を特定できないのも同じカラクリで成り立っていることが多いです。
- 今回はシステム思考による構造化・見える化を活用して課題発見解決を進める手法SaPID（さぴっど）の活用事例を紹介し、SQiP研究会のそれぞれの研究にどう活用できるのかを一緒に考えたいと思います。

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

コンテンツ

- SaPIDとは何か？
- SaPIDを活用した研究の進め方
- 　Case：ソフトウェアレビュー研究
- SaPID適用事例
- SaPIDを活用したSQiP研究会マネジメント(例)
- SaPID実践～パフォーマンス向上のポイント
- まとめ

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPIDとは何か？

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

自律・自己組織化を促進する価値共創プログラム：SaPID

Systems **a**nalysis / **S**ystems **a**pproach based **P**rocess **I**mprovement metho**D**

- 当事者自らが対象のメカニズムをシステムとして構造的に捉え、最も効果的で現実的な箇所、あるいは新しい価値創出に有効な施策・対策をうち、成果をあげていくための手法。＝**システム思考**
- チームや関係者、組織のメンバーが参画し、**デザイン思考**を併用して一緒に成長する、そして新しい価値を共創する方法を併せ持つ。

共創プログラム

チームや組織に存在する関係者全員が参画して、相互に理解しあい、それぞれの特徴や得手・不得手を融合して同じ目的を一緒によりよく達成することを促進するアプローチ群。

シャープ伝説のエンジニア　故佐々木正氏（シャープ元副社長）の言葉

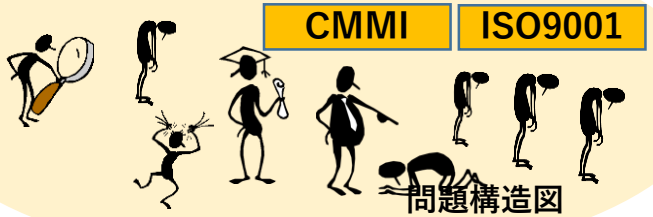
「いいかい、君たち。分からなければ聞けばいい。持っていないなら借りればいい。逆に聞かれたら教えるべきだし、持っているものは与えるべきだ。人間、一人でできることなど高が知れている。技術の世界はみんなで共に創る『共創』が肝心だ」

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPIDの主な用途

～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／やられるプロセス改善の問題



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

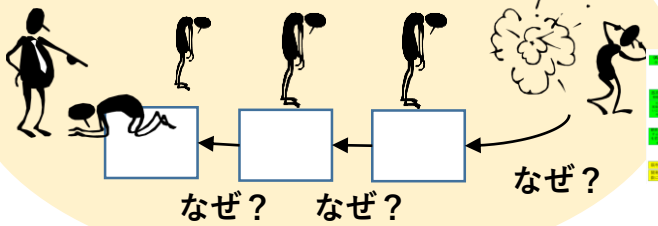
ふりかえり

FRIモデル

問題構造図
（因果関係モデル）

システム思考 ✖ デザイン思考

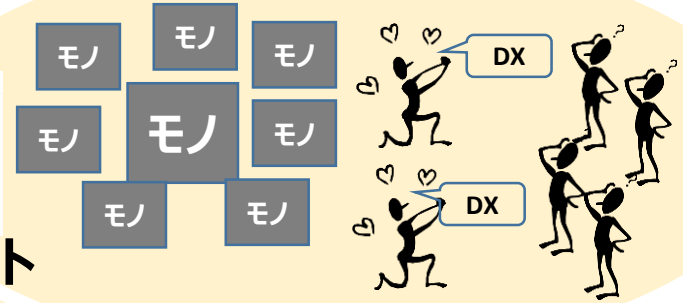
大問題しか分析しない
分析ではなく詰問になる問題



未来予想図

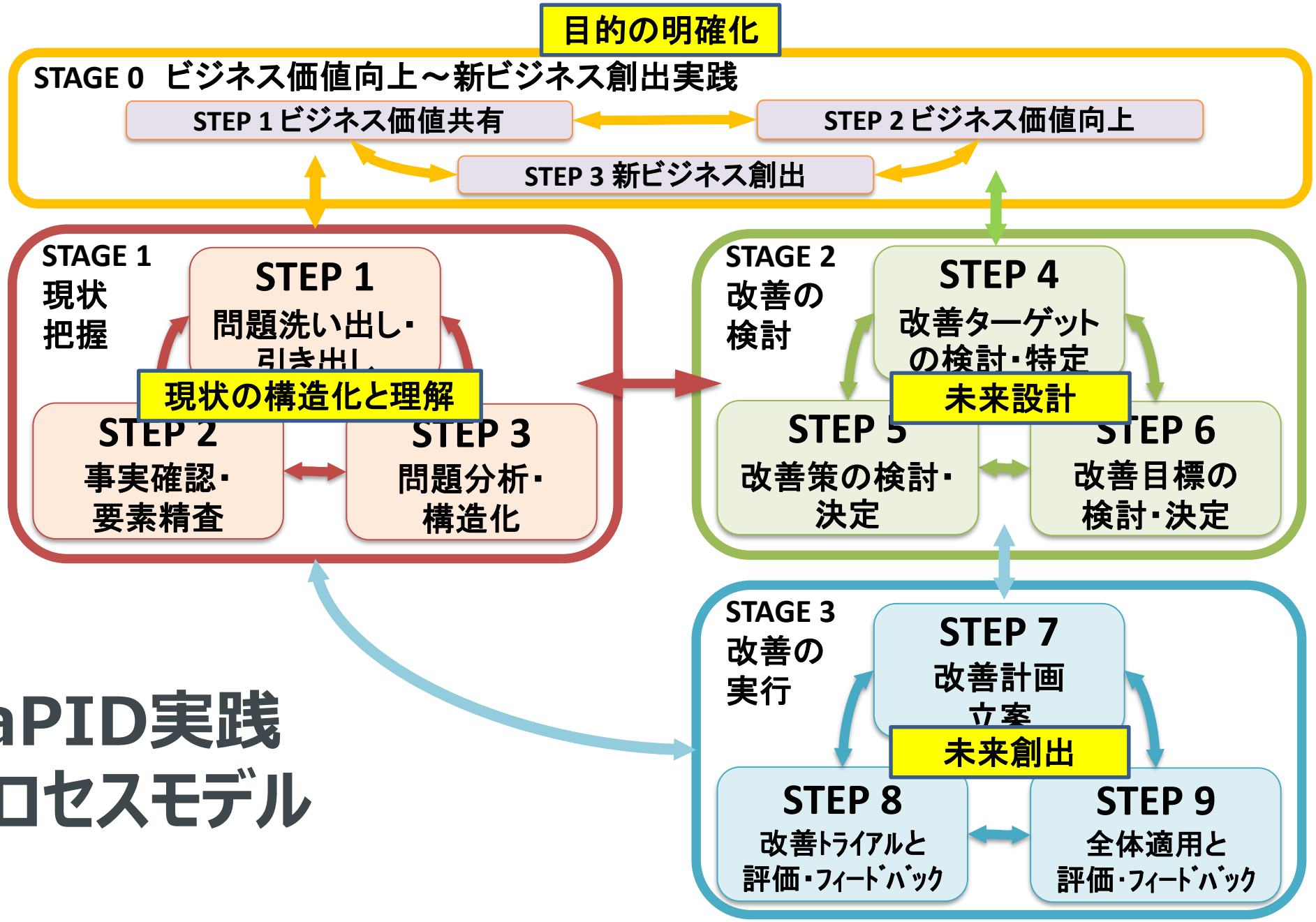
価値連鎖モデル

モノありき／コトづくりなしのエセDX実践



突然要件や仕様から始まる受け身の開発プロジェクト

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

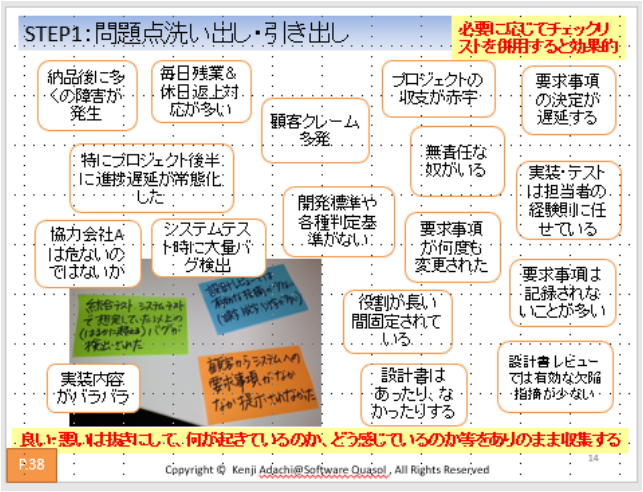


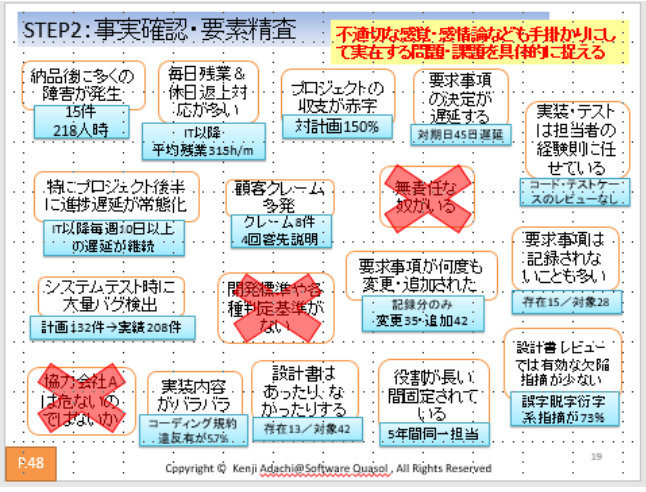
SaPID実践プロセスモデル

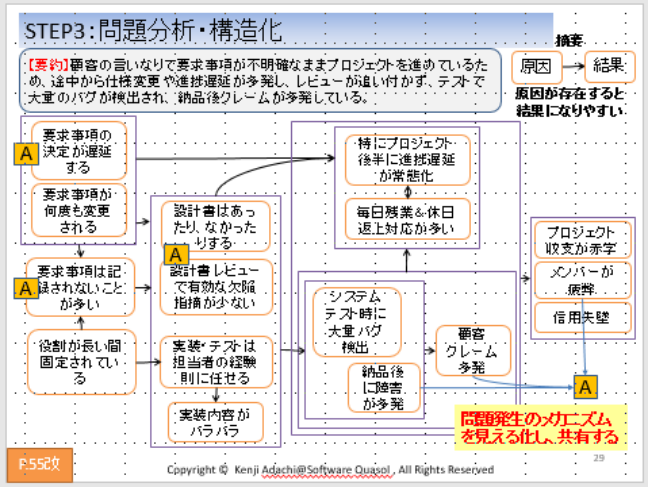
P.16

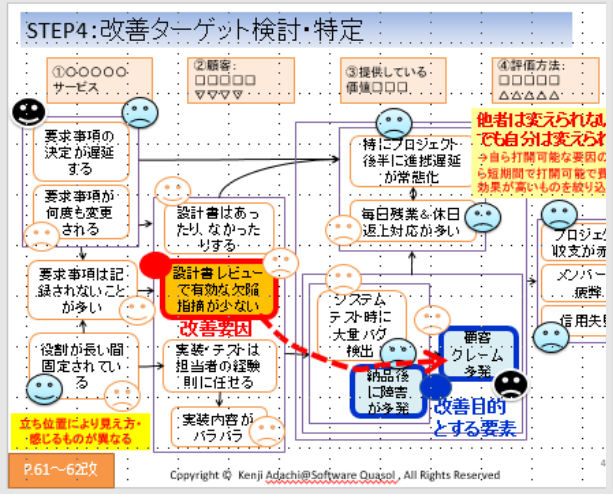
Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

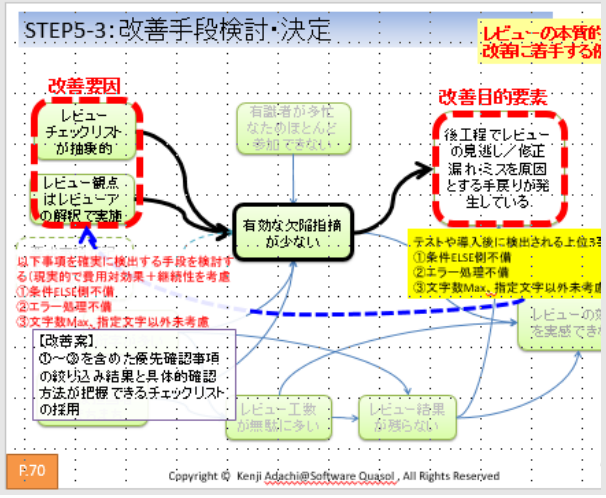
SaPIDの基本実践STEP

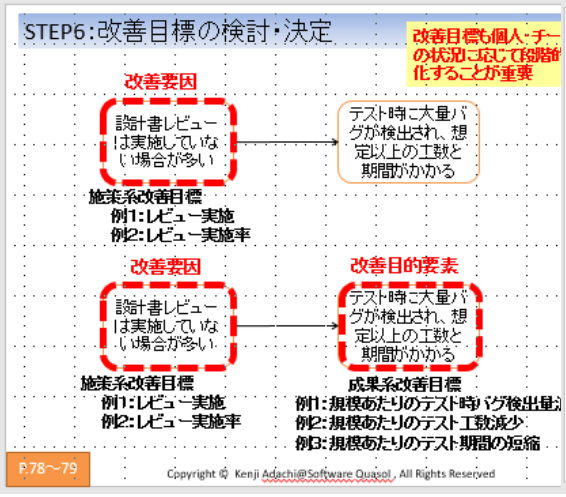


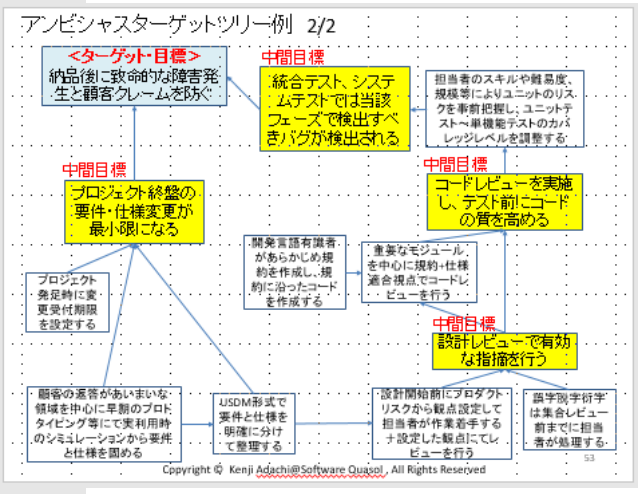








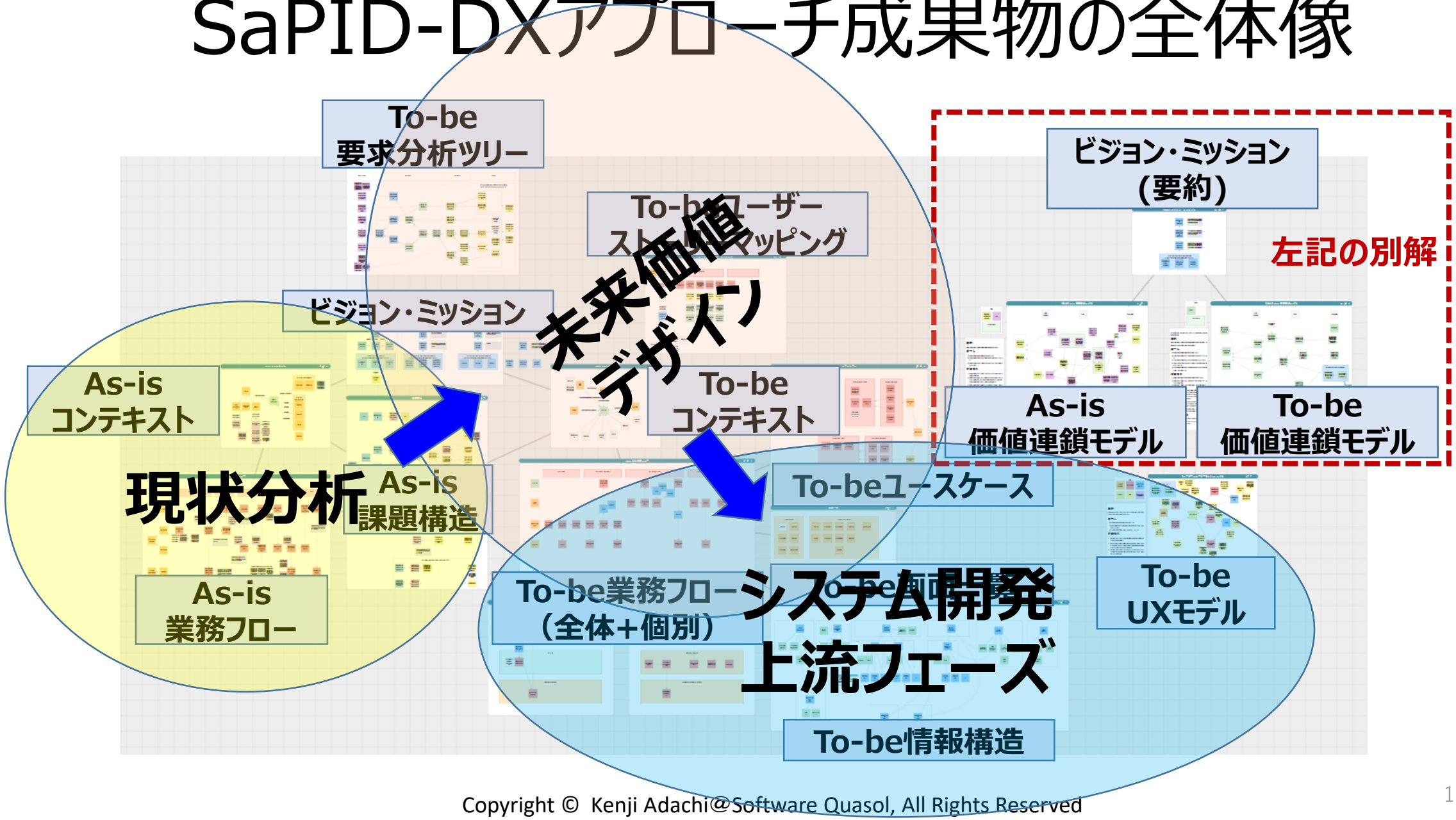




P.33-94

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPID-DXアプローチ成果物の全体像

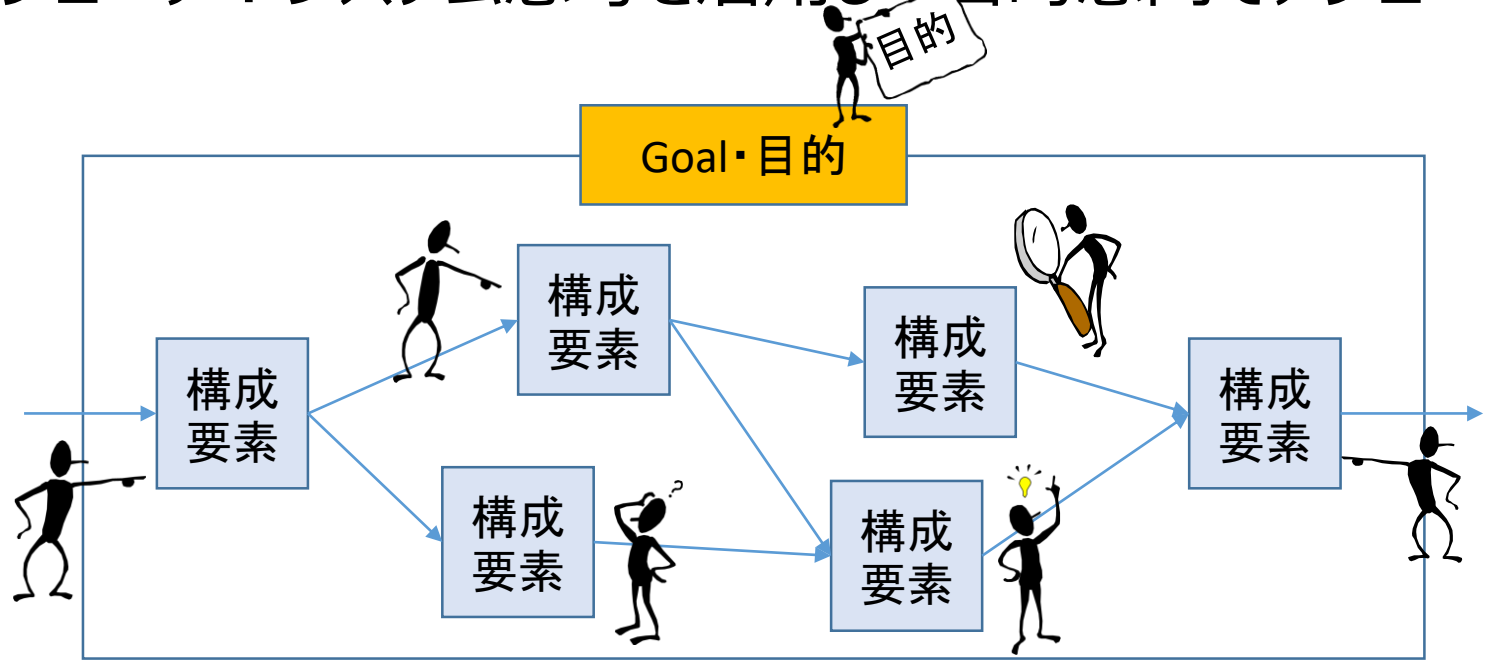


Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

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

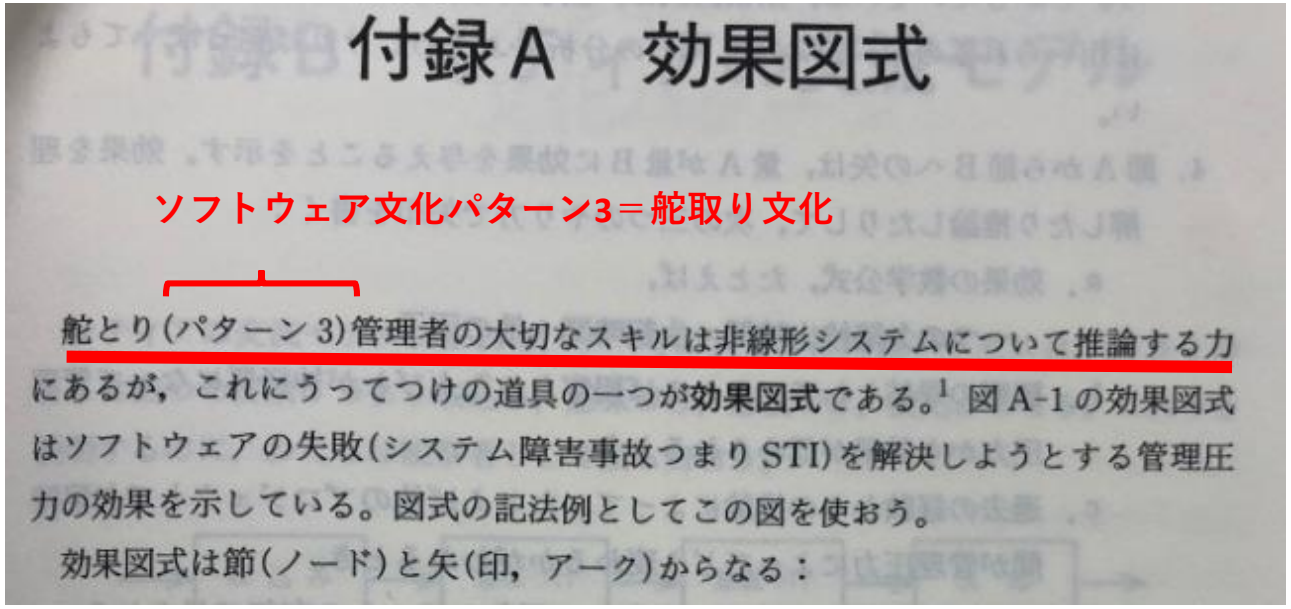
- システム思考：対象をシステムとみなして明確化する＝Systemic（俯瞰的）×Systematic（系統的）に思考する（ロジカルシンキングを含む）
- システムズアプローチ：システム思考を活用して目的志向でアプローチする



P.22

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

舵取り文化を支える
非線形システムを推論する力＝”システム思考



付録 A　効果図式

ソフトウェア文化パターン3＝舵取り文化

舵とり（パターン 3）管理者の大切なスキルは非線形システムについて推論する力にあるが，これにうってつけの道具の一つが**効果図式**である。[1] 図 A-1 の効果図式はソフトウェアの失敗（システム障害事故つまり STI）を解決しようとする管理圧力の効果を示している。図式の記法例としてこの図を使おう。

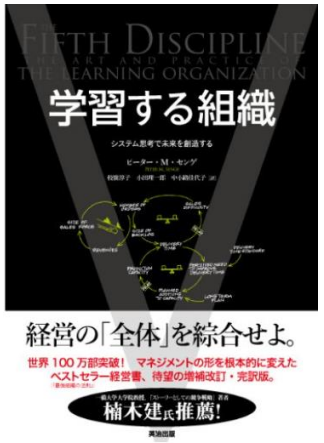
効果図式は節（ノード）と矢（印，アーク）からなる：

[ソフトウェア文化パターン]
パターン5：適合文化
パターン4：予知文化
パターン3：舵取り文化
パターン2：慣習的文化
パターン1：可変の文化
パターン0：無意識の文化

ソフトウェア文化パターン：G.M.ワインバーグ氏が工学的プロセスをどのレベルで実践する組織なのかを「管理の心的態度」で分類したもの
類似の分類には、Software CMM（プロセスの型による分類）、People CMM（組織内の人々の処遇による分類）などがある。

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

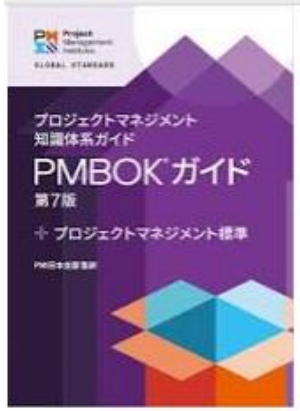
学習する組織を構築する３つの柱
ピーターMセンゲ



- 志を立てる力
  ～組織と個人のビジョン
  「現実」と「なりたい姿」のギャップの間に生まれる力を推進力に
- 複雑性の理解力
  ～システム思考
- 共創的な対話力
  ～メンタル・モデルとダイアログ

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

PMBOK 第7版
プロジェクトマネジメントの12原則



- スチュワードシップ
- 協力的なプロジェクトチームの環境を作る
- ステークホルダーを効果的に連携する
- 価値に集中する
- システムの相互作用を認識し、評価し、対応する → **システム思考**
- リーダーシップを行動で示す
- 文脈に基づいたテーラリング（カスタマイズ）
- 品質をプロセスと成果物に組み込む
- 複雑性に適応する
- リスクへの対応を最適化する
- 適応力とレジリエンスを高める
- 未来の状態を達成するために変化できる

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

デザイン思考＝みんなで探索的に創る

【5つのMode】

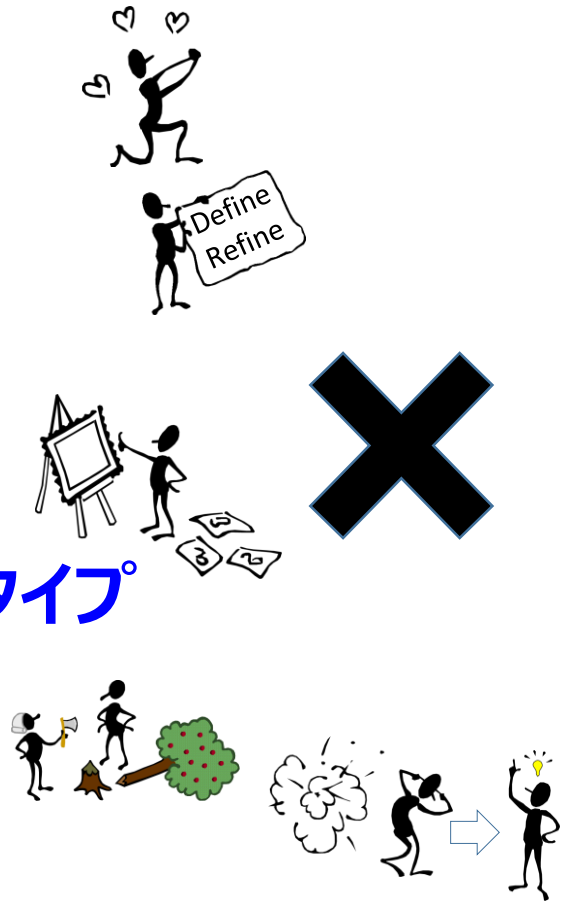
・Empathize：**共感**

・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！



【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：-the d.school bootcamp bootleg

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

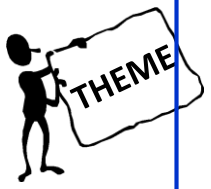
Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPID＝システム思考×デザイン思考

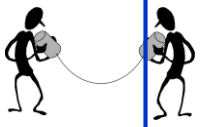
システム思考

デザイン思考

システム思考



- テーマ設定
  - 例1：〇〇開発プロジェクトの成功要因・失敗リスク
  - 例2：レビューにおける困り事・問題点



- 現状分析
  - 関係者が持つ関連情報（要素）の洗い出し
  - 構造（関係性）分析：物理モデル構築
  - 関係性構造の論理モデル化

共感・問題定義・人間中心・協力的

- 未来設計・実装・実行
  - 新たな価値連鎖モデル構築（価値設計）
  - 成長に必要な改善要素の特定と施策検討
  - [トライアル＆ふりかえり]による成果獲得の拡張

人間中心・楽観的・協力的・創造的

実験的／プロトタイプ・テスト

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPIDのコンセプト：対象全体につらぬかれた骨格となる発想や観点

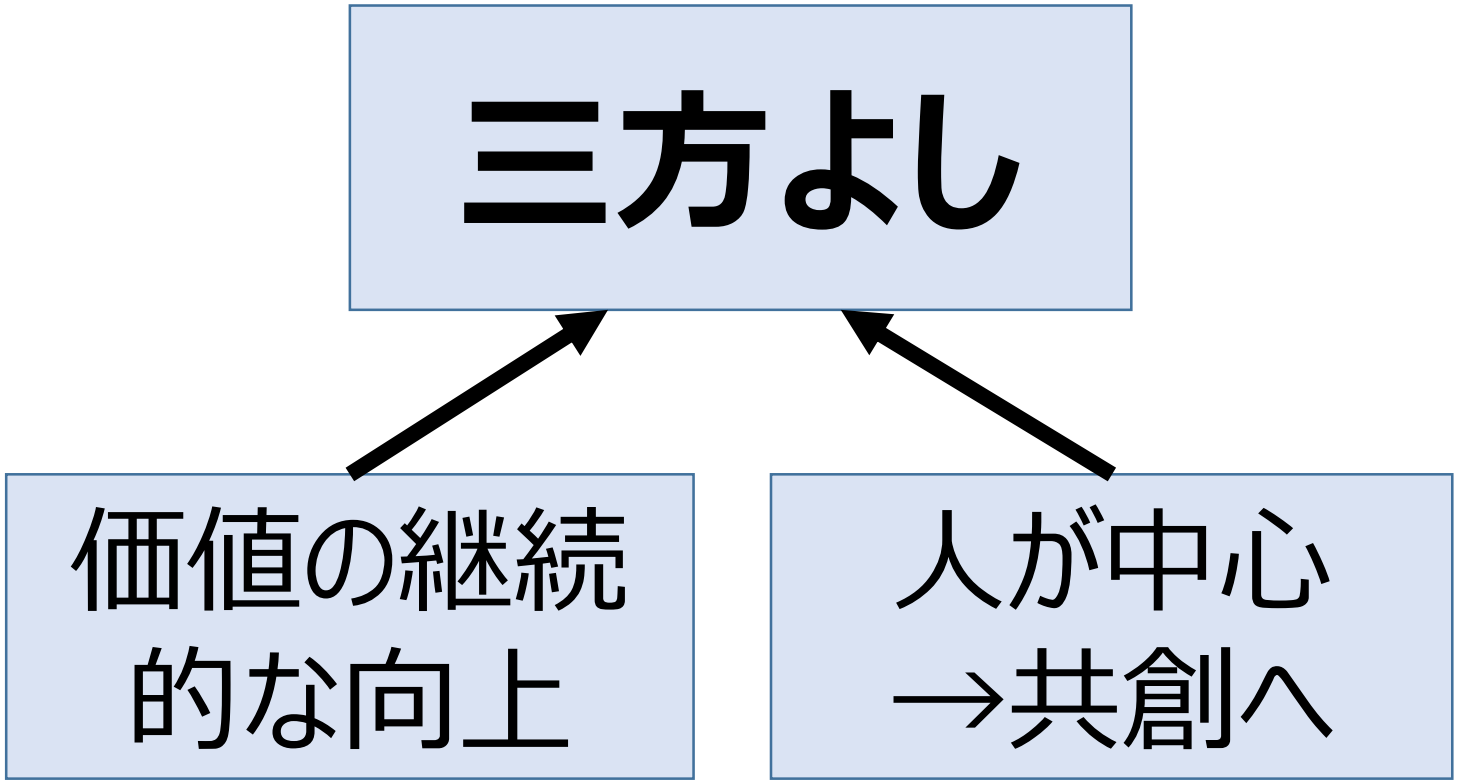
自律～自らが変化の起点になる

- 自らを律する、変えることによって結果を出す。成果を上げる。そして（その結果）周囲が変わる。

P.18

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

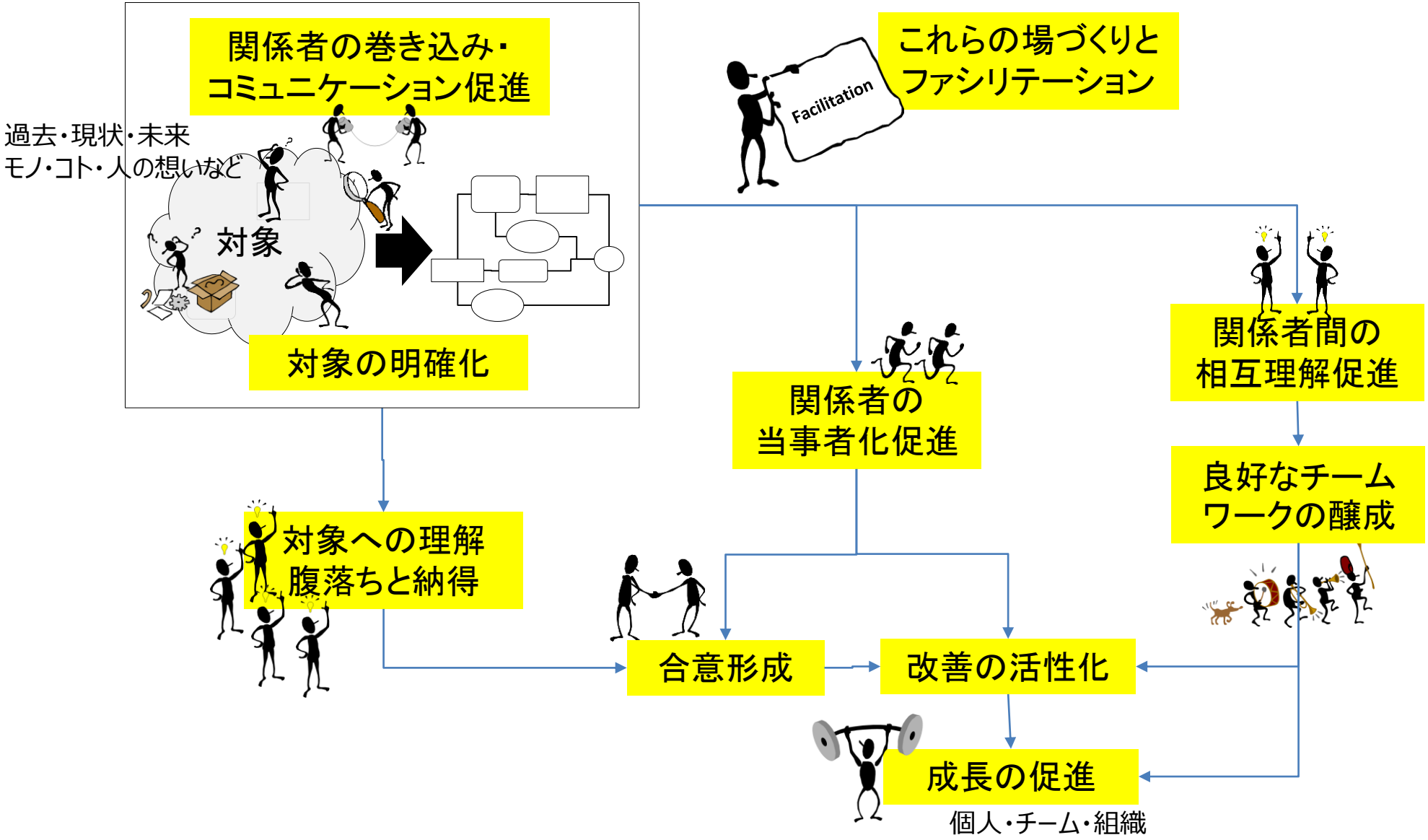
SaPIDのポリシー：物事を行うときの方針や原則



P.20・21

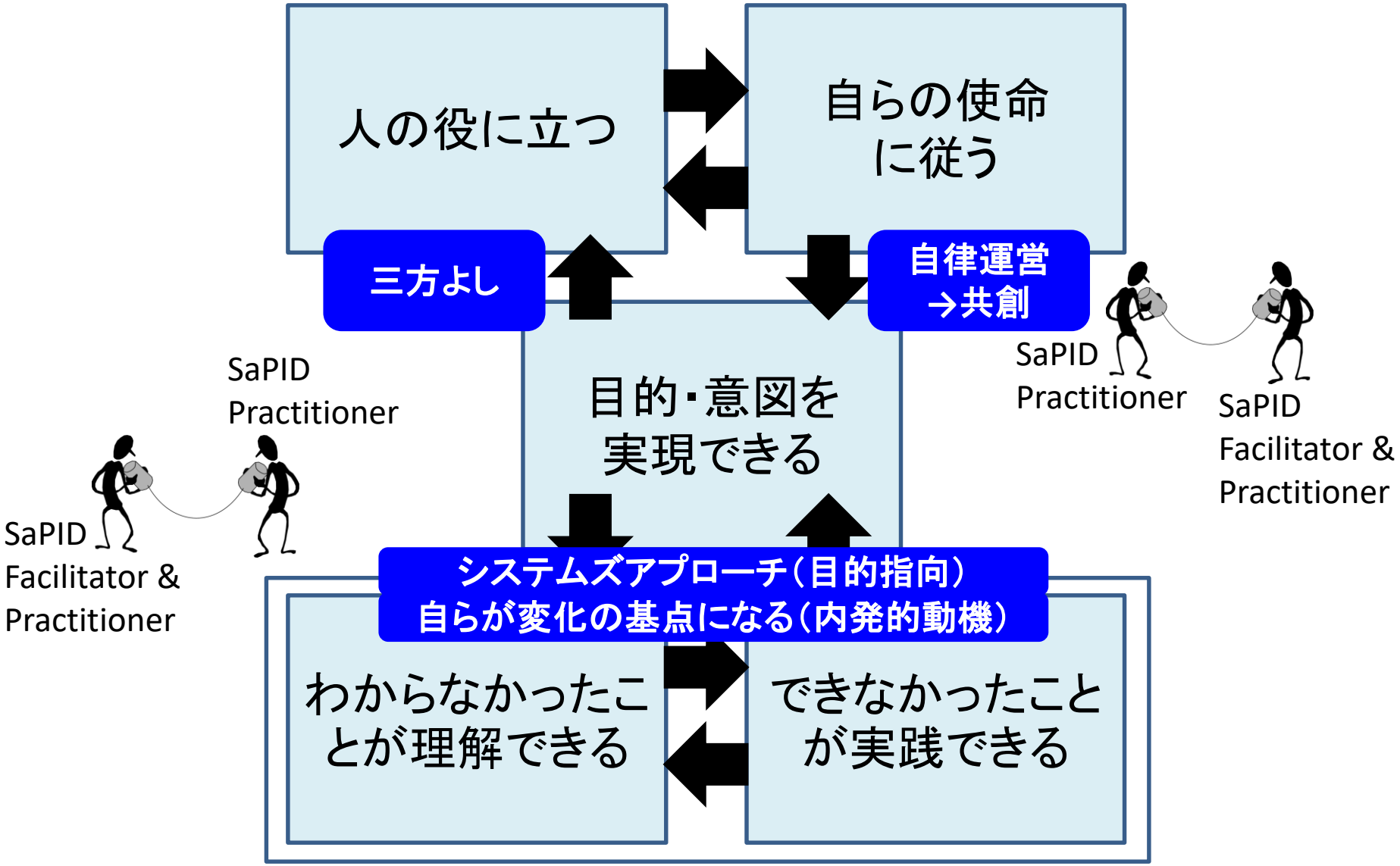
Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPIDでできること・やっていること



Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

(依存)→自律→共創(相互依存)による幸せ実現モデル



Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPIDを活用した研究の進め方
Case：ソフトウェアレビュー研究

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPIDの基本STEPに沿って実践体験

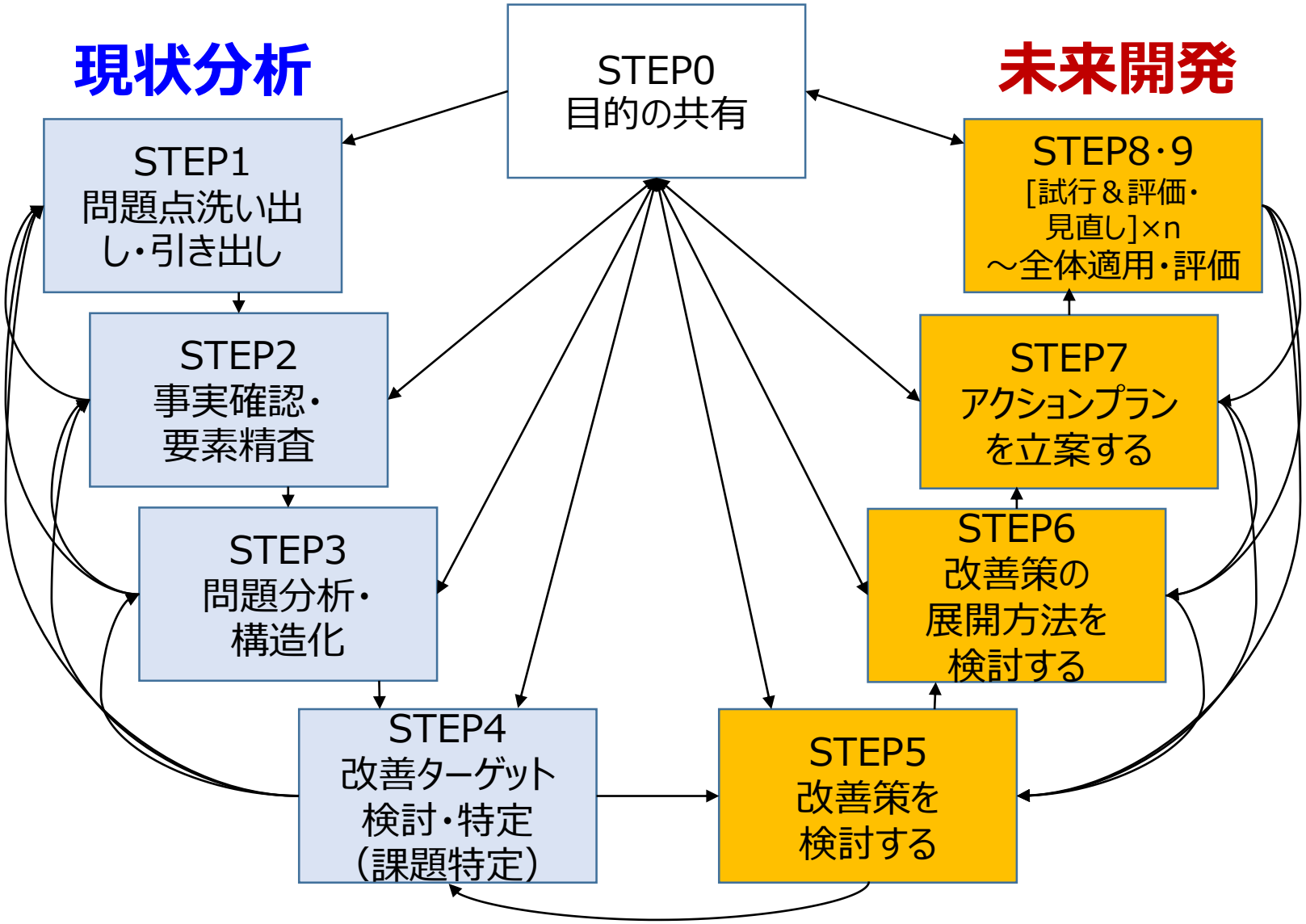
- STEP0：**「レビュー」**の目的の共有
- STEP1：**「レビュー」**の問題点洗い出し・引き出し
- STEP2：事実確認・要素精査
- STEP3：問題分析・構造化
- STEP4：改善ターゲット検討・特定（課題特定）
- STEP5：改善策の検討・決定
- STEP6：改善目標の検討・決定
- STEP7：改善計画立案
  -
  -

今回はこの「現状分析」を一緒にやってみましょう！

「レビュー」の部分を別の対象に変えればそのまま活用できます。

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPIDアプローチ全体像



P.16

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

STEP0：「レビュー」の目的の共有

①自らの仕事（業務）と役割

〇〇〇〇〇サービス

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？

この対応過程で自らのミッションやポリシーが明確か？本来は何なのか？が明らかになるのが理想

P.33～37

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

STEP1：「レビュー」の問題点洗い出し・引き出し

自由に1件1付箋に記述する

- 当事者意識が少ない
- 分析（区分選択など）は面倒
- 実施側や受入時に指摘される
- 細かくみると時間がかかる
- レビュー記録作成に時間がかかる
- チェックリストが形骸化する
- 単なる儀式になっている
- かけた工数に対して指摘数が少なく些末な指摘しかできない
- レビュー指摘件数等の指標が蓄積できていない
- チェックリストでレビューを行うがレビュー漏れが発生する
- 業務・システム詳細につっこんだレビューが出来ない
- レビュアーによって言うことが異なる
- レビュー実施工数に対してお客様の理解が得られない
- 後工程で欠陥が発見される
- レビューにかかる工数（費用）を確保できない
- 有識者のスケジュール調整に
- レビューを繰り
- うちに落としど
- が見つからず
- が増える
- 味は
- 疑問に
- ーの重
- 要とが拡がらない
- ないことがある
- 誤字・脱字等体裁チェックの結果ばかりになる
- に目が奪われ、内容チェックに入れない
- レビューポイントが明確でない
- レビュー工数が不足する
- レビュー対象が膨大
- 専門的な知識を有するレビューアの育成がなかなか進まない
- 指標値の収集・分析活動が組織として定着していない
- 指標値は少数の大規模件名の結果から作成されたもの
- 小規模件名に対する妥当性が不明
- 責任感がない・自己チェックが甘い
- レビューの指摘が曖昧で対応に苦慮する
- 指標値外れに対する品質の妥当性確認が上手くいかない
- 全指摘事項を組込むと煩雑なので採用、不採用を判断するがどこでけりをつけていいのか分からないときがある
- レビュー漏れがないかの確信が持てない
- レビューアがレビュー対象の業務・システム詳細理解度が低い
- レビュー結果を記述する掲示板に記載がないのは見ているのかどうかがわからない
- レビューに耐えられない成果品をレビューする
- 仔細な指摘が多い
- レビュー観点が定まっていない
- レビューが進捗のボトルネックになりやすい

P.38

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

STEP2：事実確認・要素精査

- **事実に立脚したアプローチが必要**

  事実ではない情報は除外する

ただし、SaPIDでは「〇〇で嫌な思いをしている」とか「□□が面白くない」などの感情面の状況は積極的に収集する→その背後に存在する事象を明確にするきっかけにする

- **問題点を具体的にありのまま把握する**

  不明点や疑問点がある→質問して解消する

  複数の内容が混在している→1件ずつ分解する

  抽象的な内容→具体的内容に変換

  問題点もどき表現や内容→問題点に変換

P.48

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

文章表現の原則・禁則

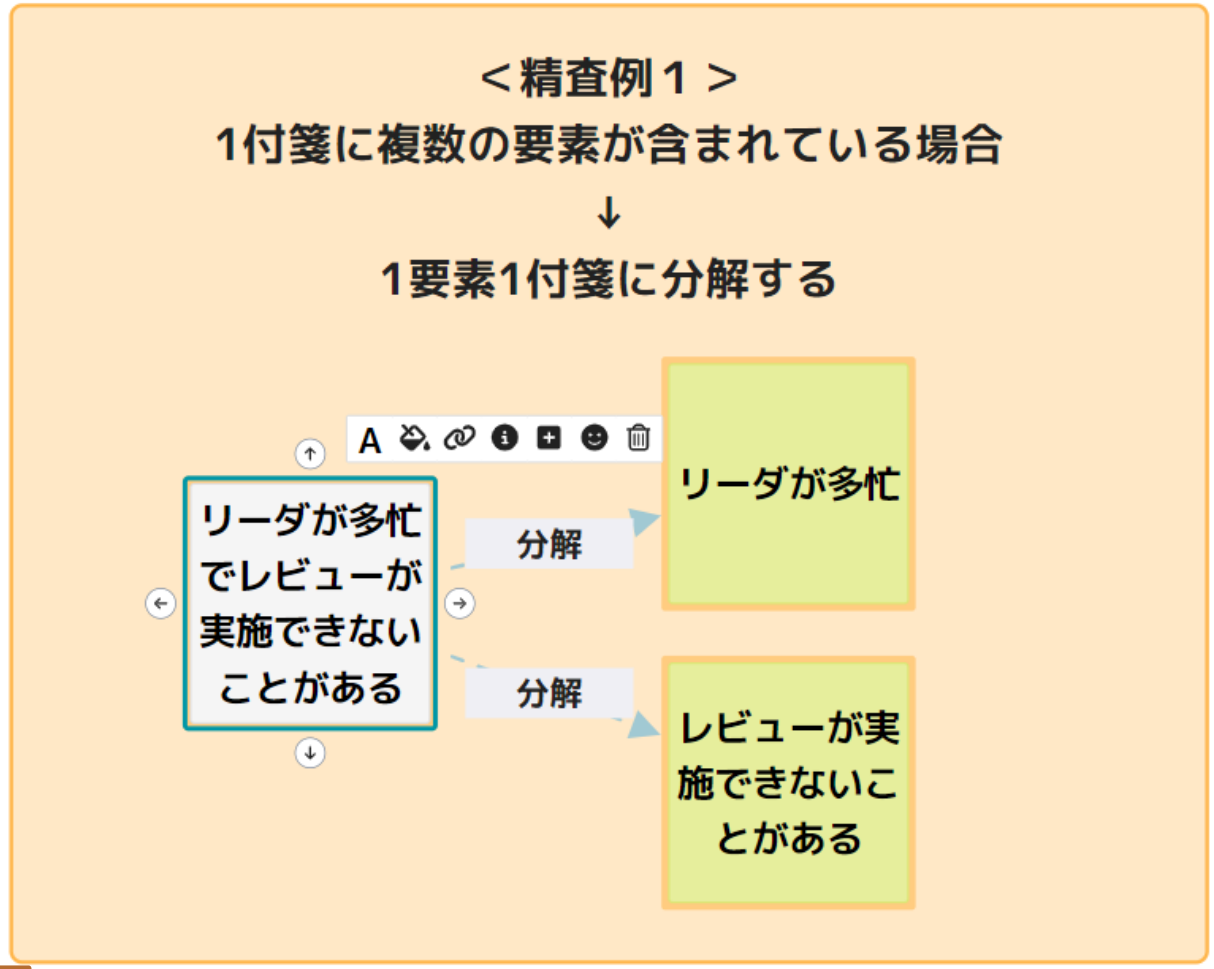
文章表現の原則	内容
事実準拠の原則	事実に即して記載する。
断定・推測区分の原則	推測を含める場合は事実と分けてはっきりと記載する。
個性化の原則	決まり文句や流行語、一般論的な表現を避け、実際に存在する個別事項を記載する。
共通理解の原則	訴えたいことがそのまま関係者に理解されるように表現する。
具体化の原則	抽象的な表現を避け、どのような状態や結果なのかを具体的に把握できるように記載する。
一文一義の原則	一つの文に一つの内容を記載する。
簡潔性の原則	余計な修飾語や冗長な説明を削り落として簡潔に記載する。

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。

P.52

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

複数の内容が混在している→1件ずつ分解する



発生している事象が
(1)どうして発生したのか
(2)その事象がどのような結果につながっているのか
をSTEP3で系統的に分析するために個別要素に分解しておく

P.52

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

抽象的な内容→具体的内容に変換

＜精査例2＞
抽象的でよくわからない内容の場合
↓
いつ、何が、どのように起きているのか
具体的に書き換える

リーダが多忙 → 具体的に書き換え → リーダの稼働が継続して36協定違反の状態

例.

レビューの
生産性が悪い

そう感じた具体的エピソードを教えて！

12頁のレビュー対象を半日かけてレビューしている

軽微な欠陥ばかり指摘される

あとになって見逃した重大欠陥が明るみになることが多い

P.52

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

問題点もどき表現や内容→問題点に変換

＜精査例3＞
〇〇不足／がない・△△が欲しい・□□に弱いなど
↓
それがない、不足していることで
発生している事象に書き換える

レビューチェックリストがない → 書き換え → レビューで重大な欠陥の見逃しが多い

例.

品質基準
がない

それがない（あるいは不足した）ことで何が起きてる？

レビュー完了判定が主観で実施される

完了判定に自信がない

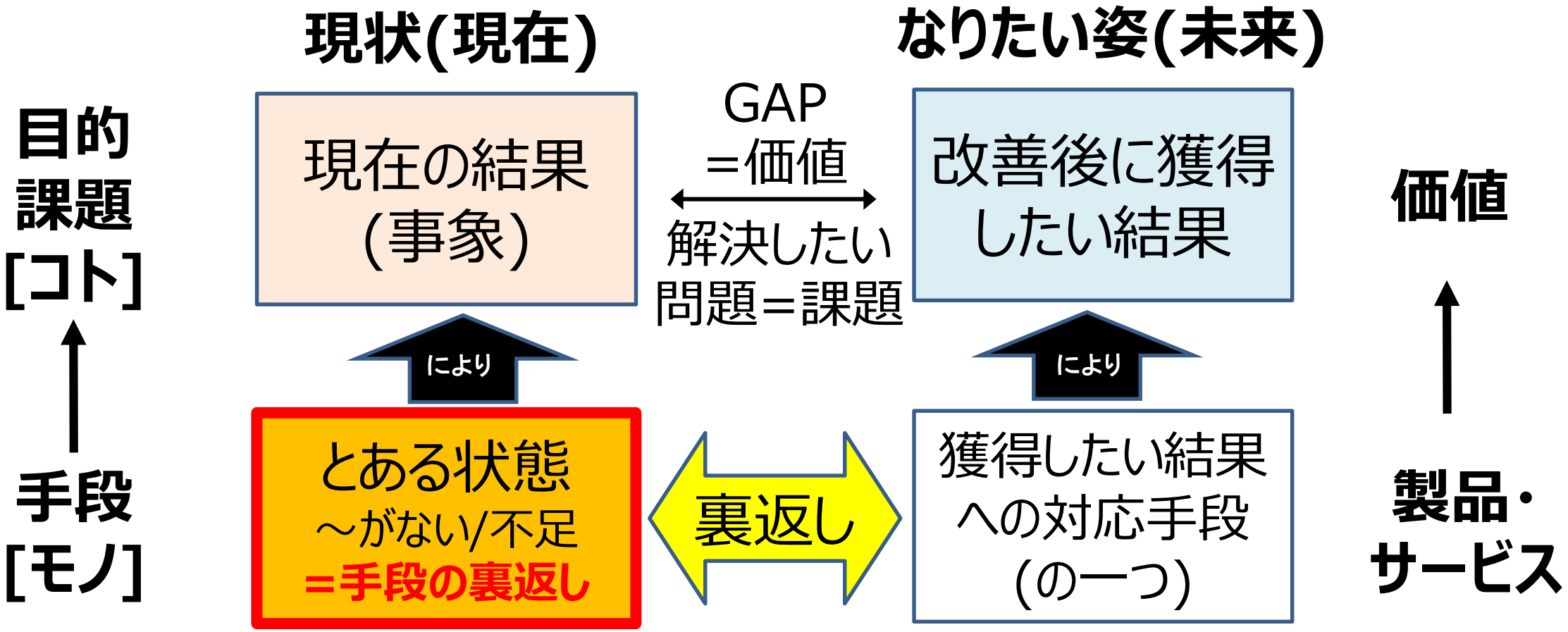
後でレビュー漏れに起因する障害発生が多い

P.52

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

「手段の裏返し」を問題点や困り事として捉える人は多い

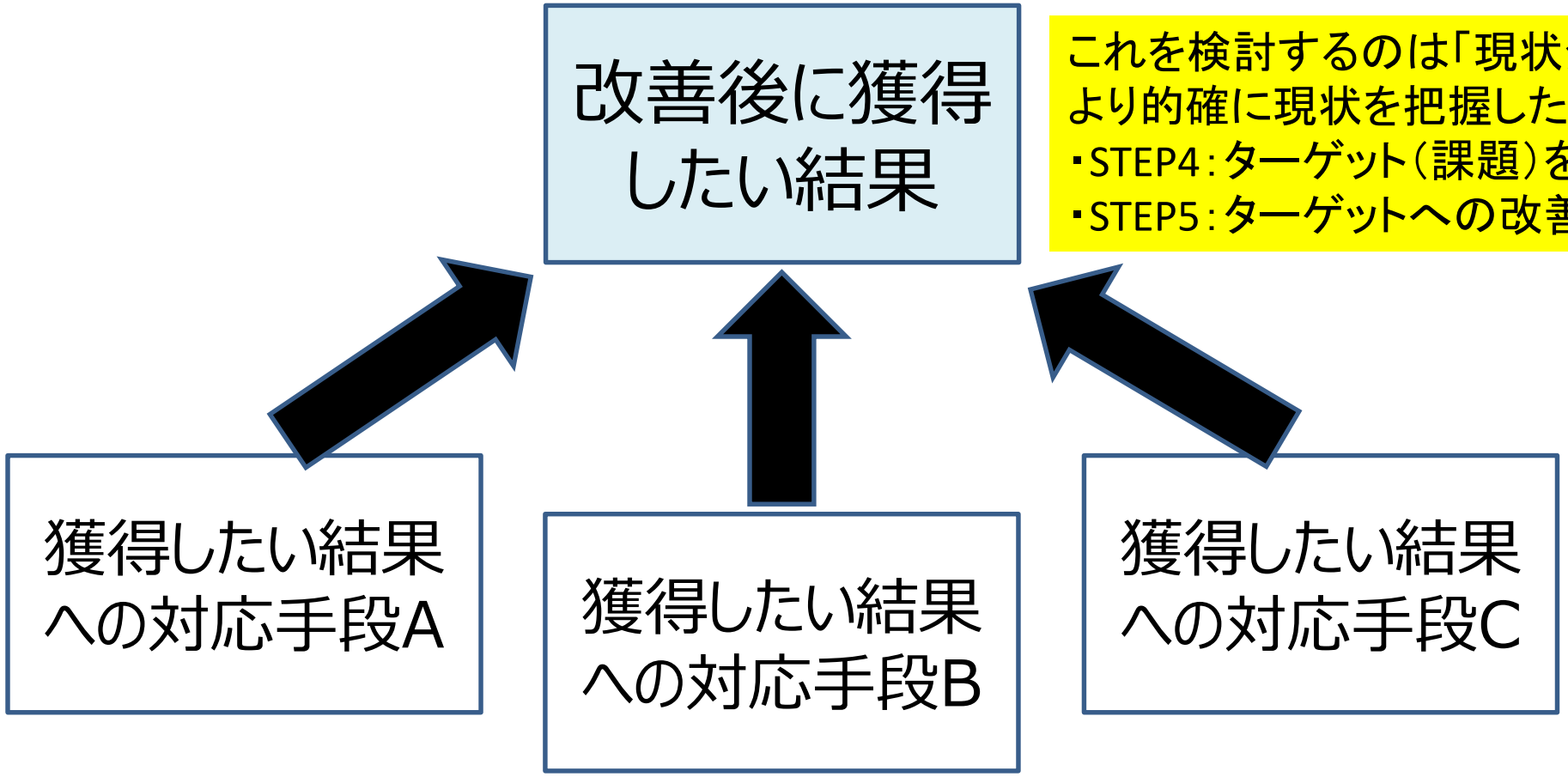
その情報から課題や目的を明らかにする



Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

ある結果を導く手段は通常複数ある

複数の選択肢の中からより適切なものを選ぶ必要がある



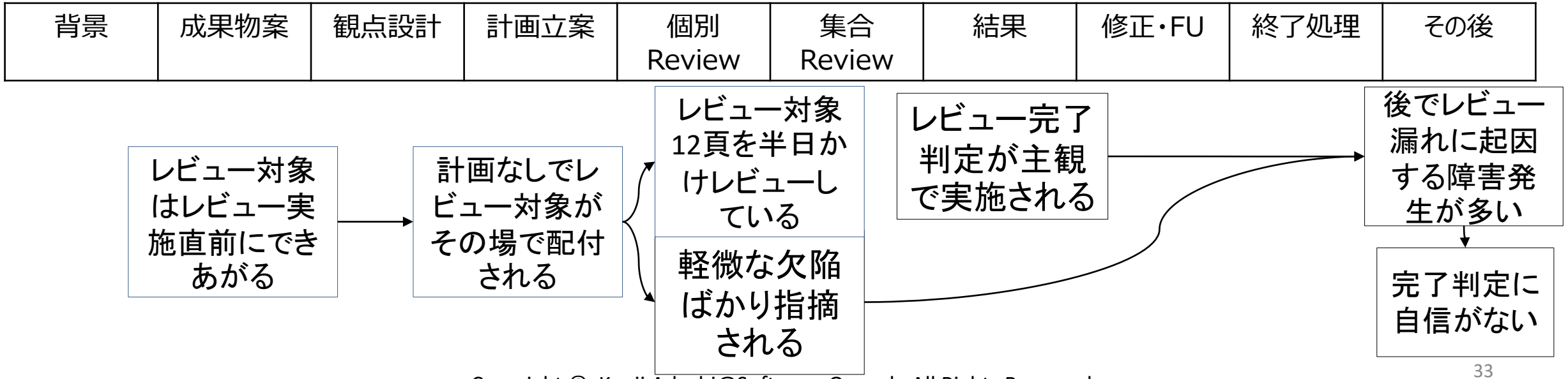
これを検討するのは「現状分析」後
より的確に現状を把握したうえで、
・STEP4：ターゲット（課題）を決める
・STEP5：ターゲットへの改善策を検討する

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

STEP3：問題分析・構造化

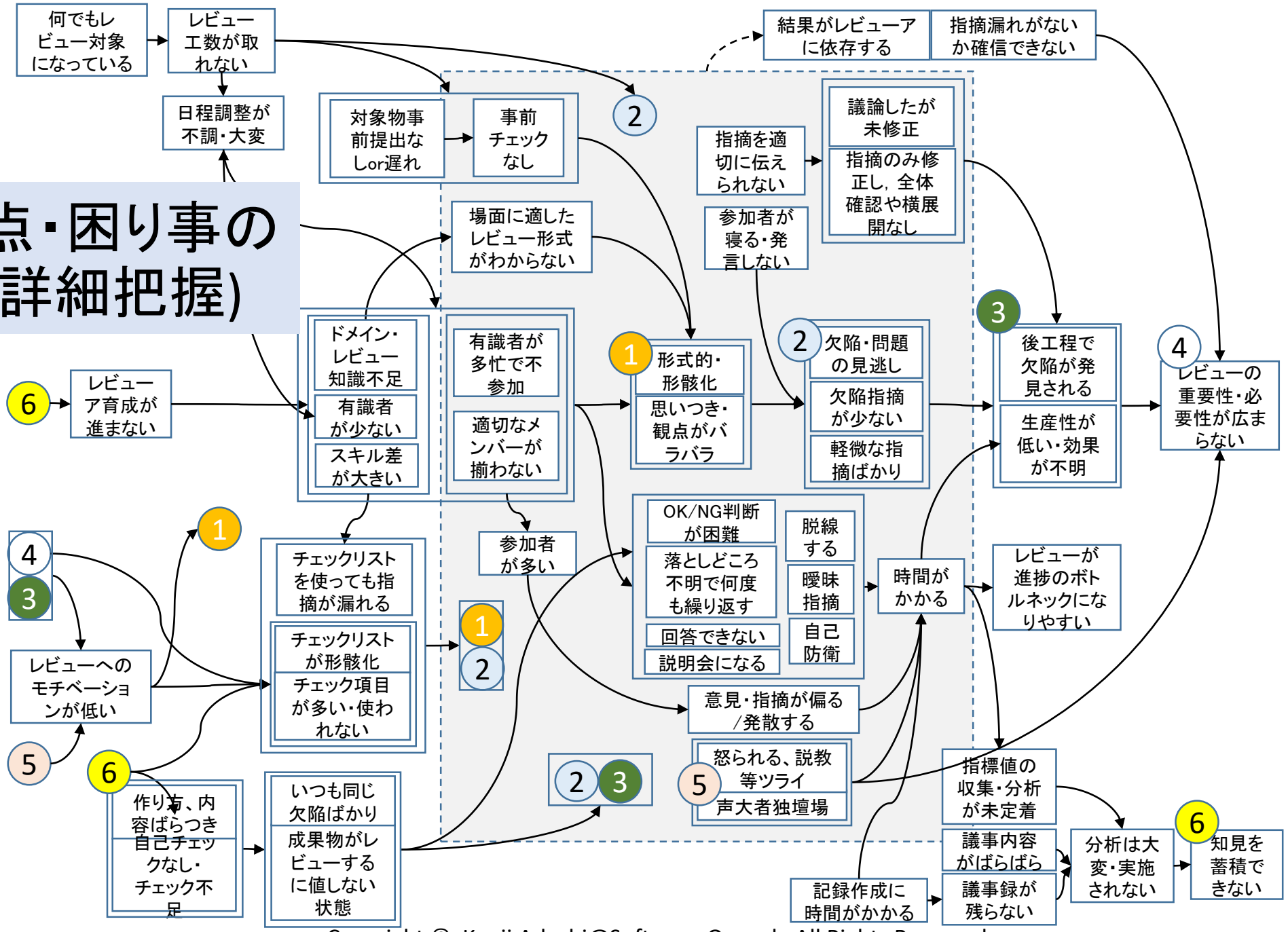
- 事実として抽出し、具体的に把握した問題点や困り事の関係性を分析して構造化する。

  ～ライフサイクル上にそれぞれの要素を配置し、因果関係がある要素同士を矢印で結びながらカラクリを明らかにする。



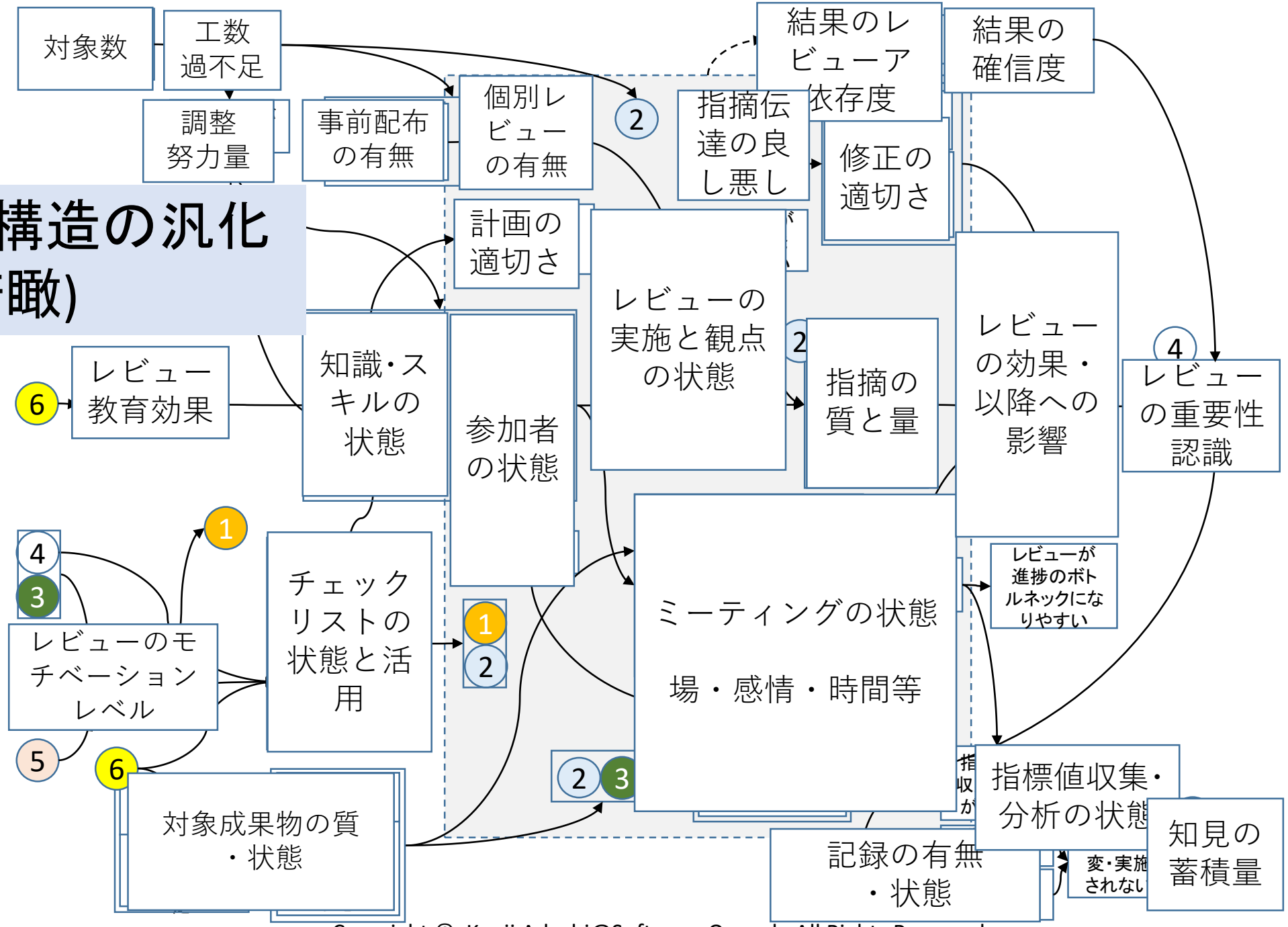
Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

■問題点・困り事の構造化(詳細把握)



Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

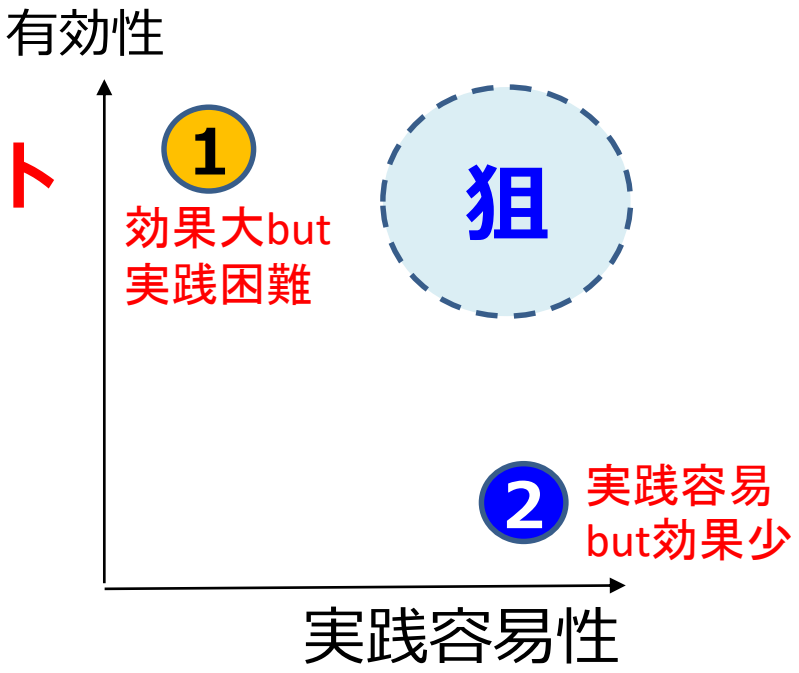
■問題構造の汎化(全体俯瞰)



Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

STEP4：改善ターゲット検討・特定（課題特定）

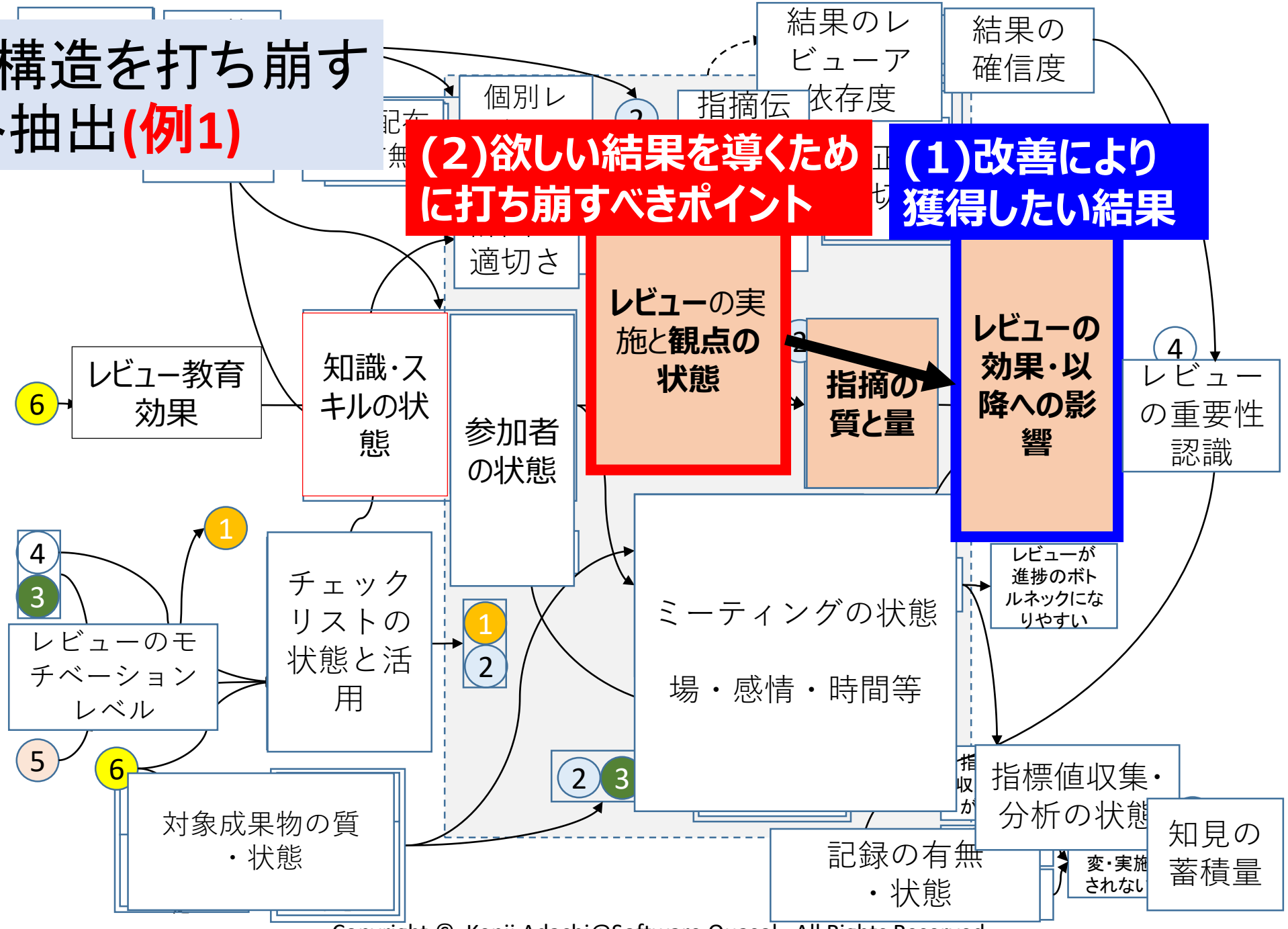
- 構造化された全体像から、改善ターゲットを特定する。
  - 改善ターゲット
    (1)改善により獲得したい結果
    (2)欲しい結果を導くために打ち崩すべきポイント



- (2)を検討する際に考慮すべきポイント
  - 有効性×実践容易性

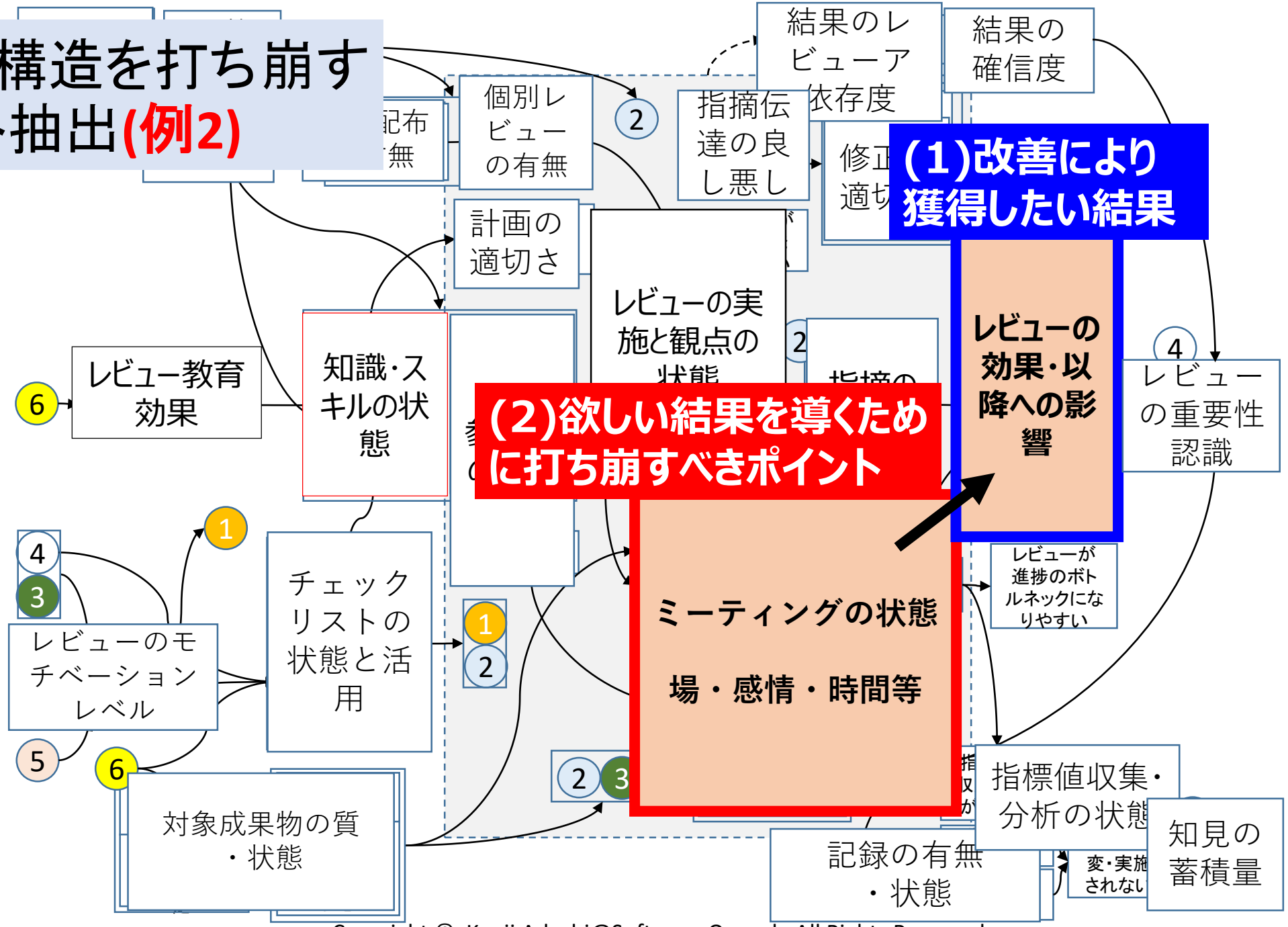
Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

■問題構造を打ち崩すポイント抽出(例1)



Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

■問題構造を打ち崩すポイント抽出(例2)



Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

構造を共有するメリット／デメリット

- **問題、困り事を一つひとつ吟味、対処する必要がなくなる。**
  リスク一覧、課題一覧の場合、1件単位に吟味、対処するので煩雑＆無駄も多い
- **関係者がみなで検討・議論し、納得・合意しやすい。**
  誰が言ったのか？ではなく、**何が適切か？どうして適切か？**で判断できる
- **複数の選択肢、解決シナリオが存在するため、現状によりフィットするものを選ぶことができる。**
- **（分析力が備わると）モノゴトの意味把握、パターン認知が進む**
  その後の活動で[現状分析→先読み＋リスク対策]が自然とできるようになる

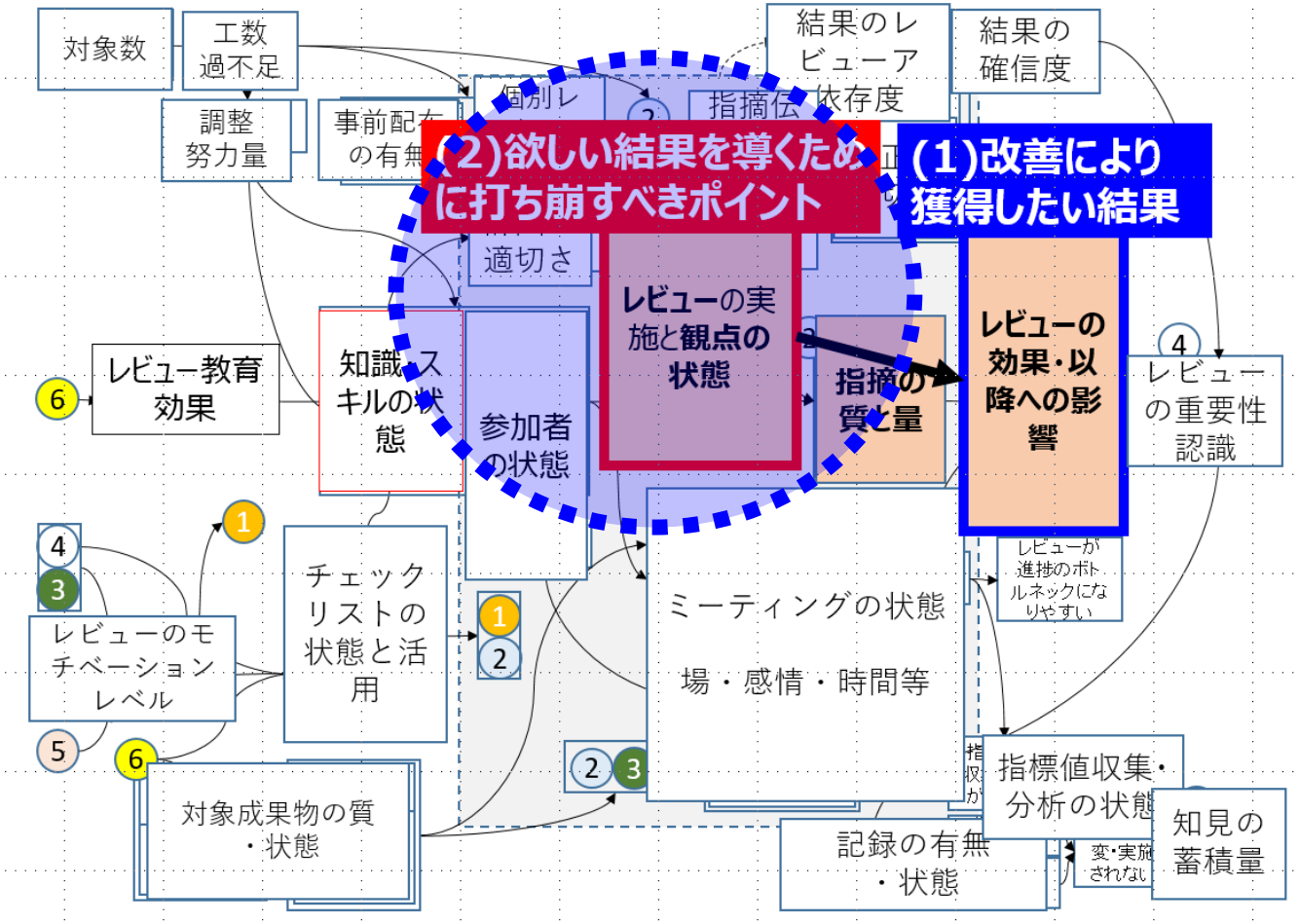
デメリット

目先で手間がかかる・関係者が揃って分析する場づくり、統制の難易度が高い　など

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

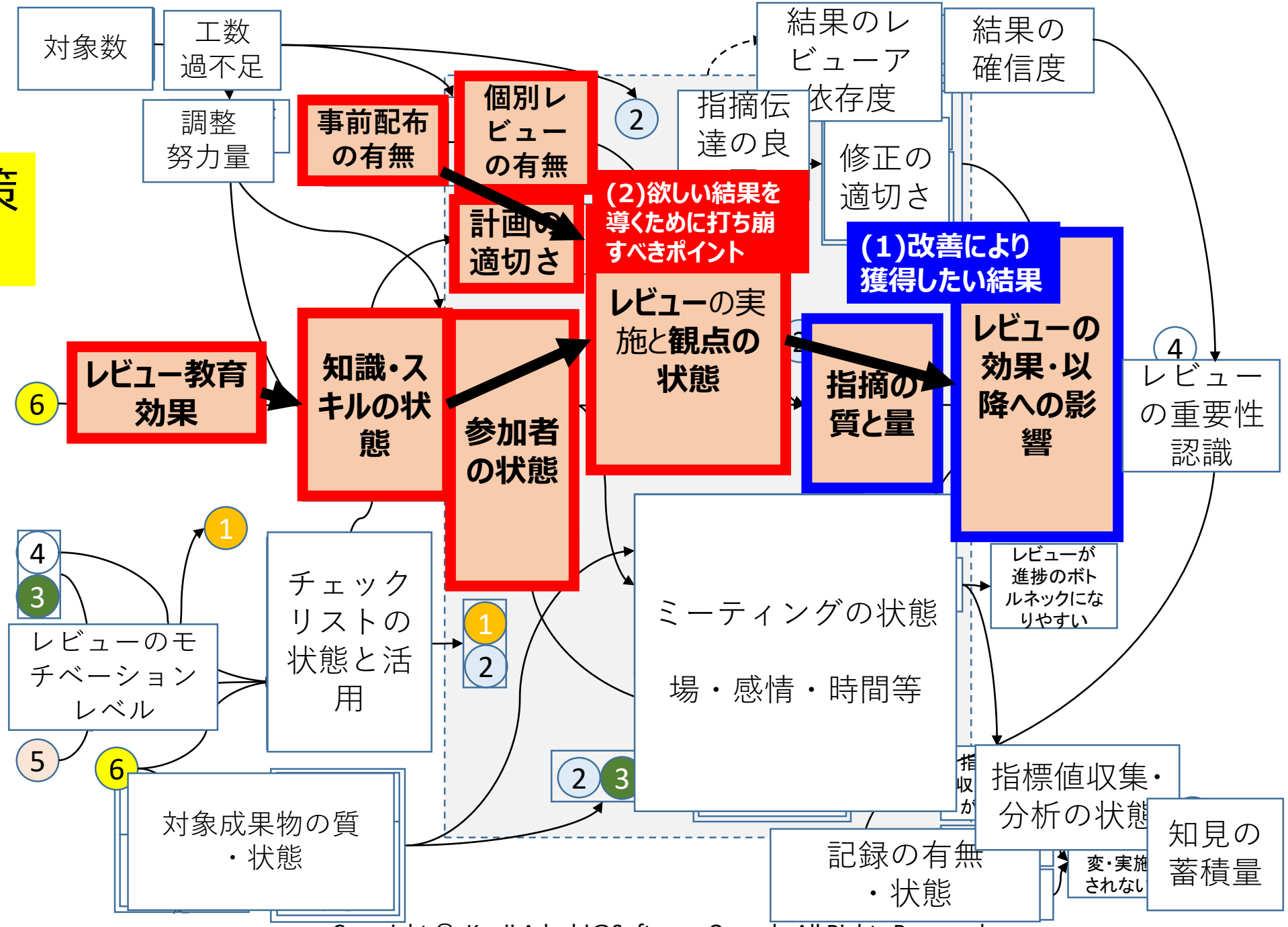
STEP5：改善策を検討する

(2)欲しい結果を導くために打ち崩す策を具体化する



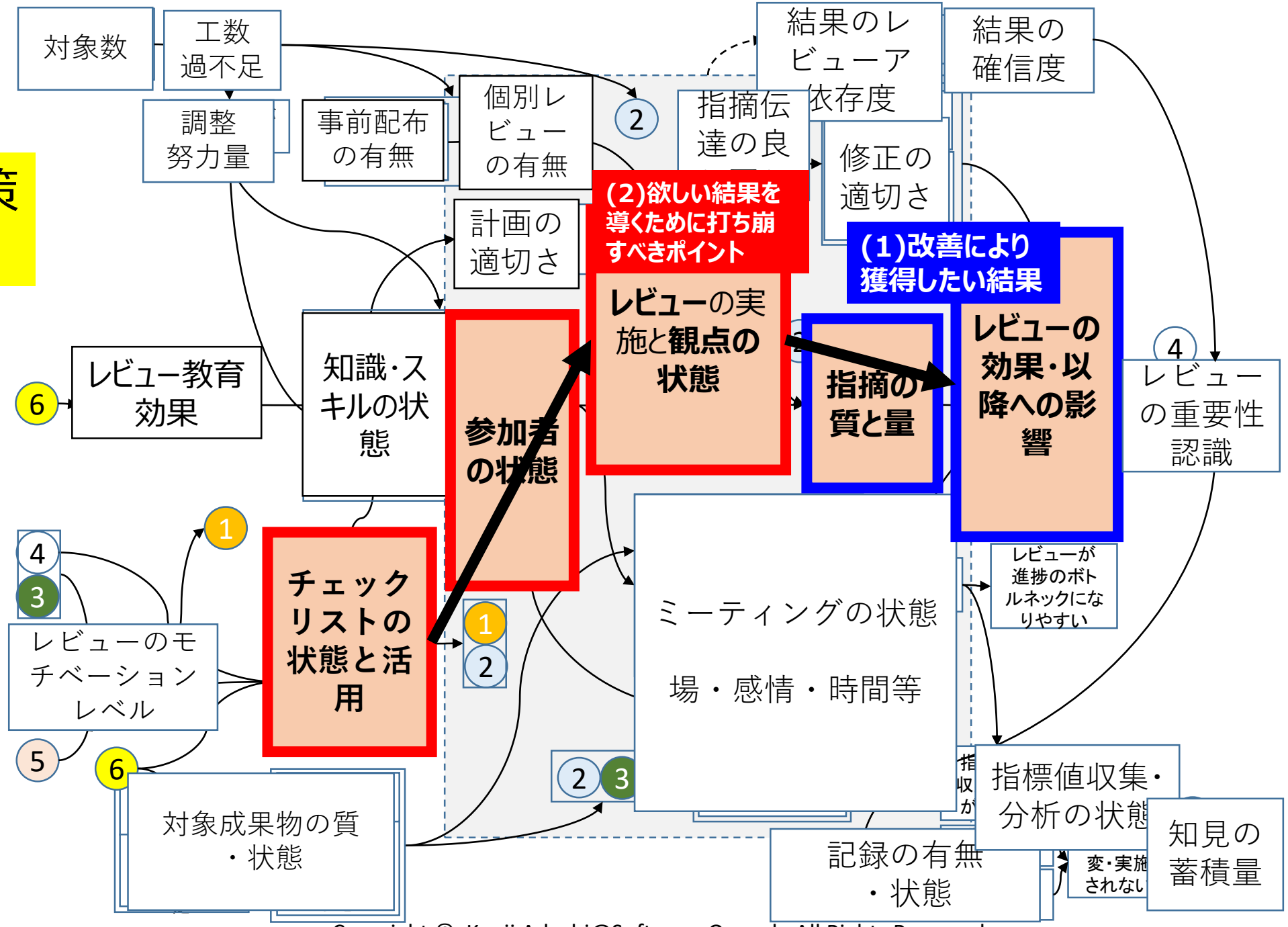
Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

改善策案1



Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

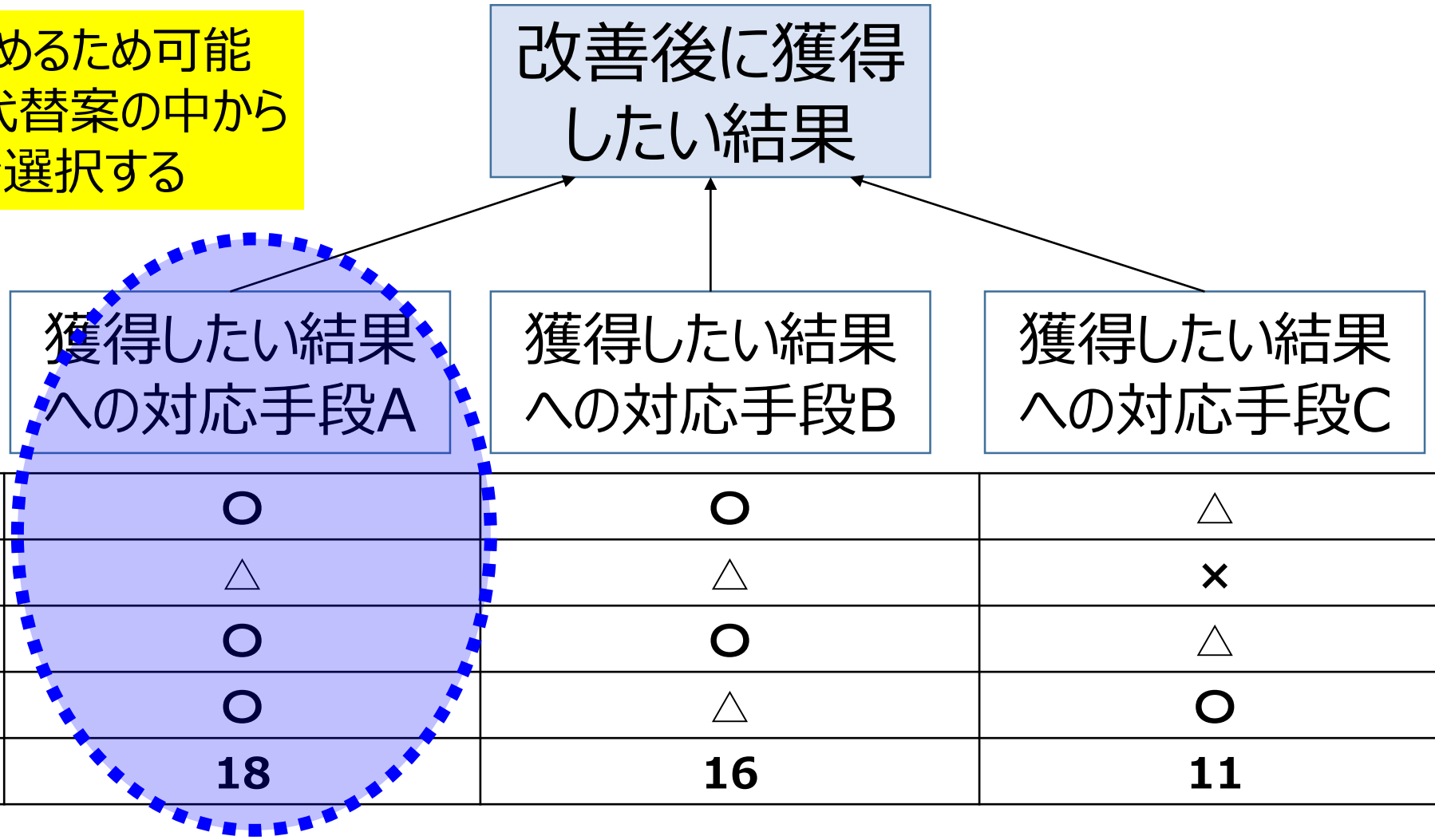
改善策案2



Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

複数の改善策から選択する

成功確率を高めるため可能な限り複数の代替案の中からより適切な策を選択する



評価指標1	○	○	△
評価指標2	△	△	×
・	○	○	△
評価指標n	○	△	○
総評	**18**	**16**	**11**

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

さらなる具体化＝ソリューション検討

改善策案1

レビュー観点導出パターン探求→見える化
レビュー有識者の思考パターン解析

ID	技法の概要
M1	作成者コンテキストからレビュー観点設定
M2	利害関係者の関心事からレビュー観点設定
M3	プロダクトリスクからレビュー観点設定
M4	対象成果物に求められる事項から確認方法設定
M5	対象成果物の典型的な欠陥から確認方法設定
M6	利用シナリオ＋状態遷移図に沿ってレビュー実施

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

さらなる具体化＝ソリューション検討

改善策
案1

レビュー観点導出パターン例
～アプローチの効果を最大化するためのコア技術

(M1)作成者コンテキストから観点導出

着目した兆候	兆候からの仮説	想定される欠陥	レビュー観点
例 当wordファイルの当初書き込み日時は、5年前の3/28 AM2:58。	過去のプロジェクト成果物を流用した可能性大	□部分流用した結果、今回の要件に合わない状態になっている □旧成果物の残骸がそのまま残されている	今回の内容には不必要な/不適切な旧成果物の残骸はないか?

(M2)利害関係者の関心事から観点導出

利害関係者	対象システムにおける主な活動	システムへの関心事（期待・疑問・懸念等）	必要なレビュー観点・対象
例．システム設計担当者	当システムの基本設計を担当	システム設計に必要な情報が漏れなく入手できるか？	システムに求められる要件が理由や目的、制約事項と共に明示されているか？

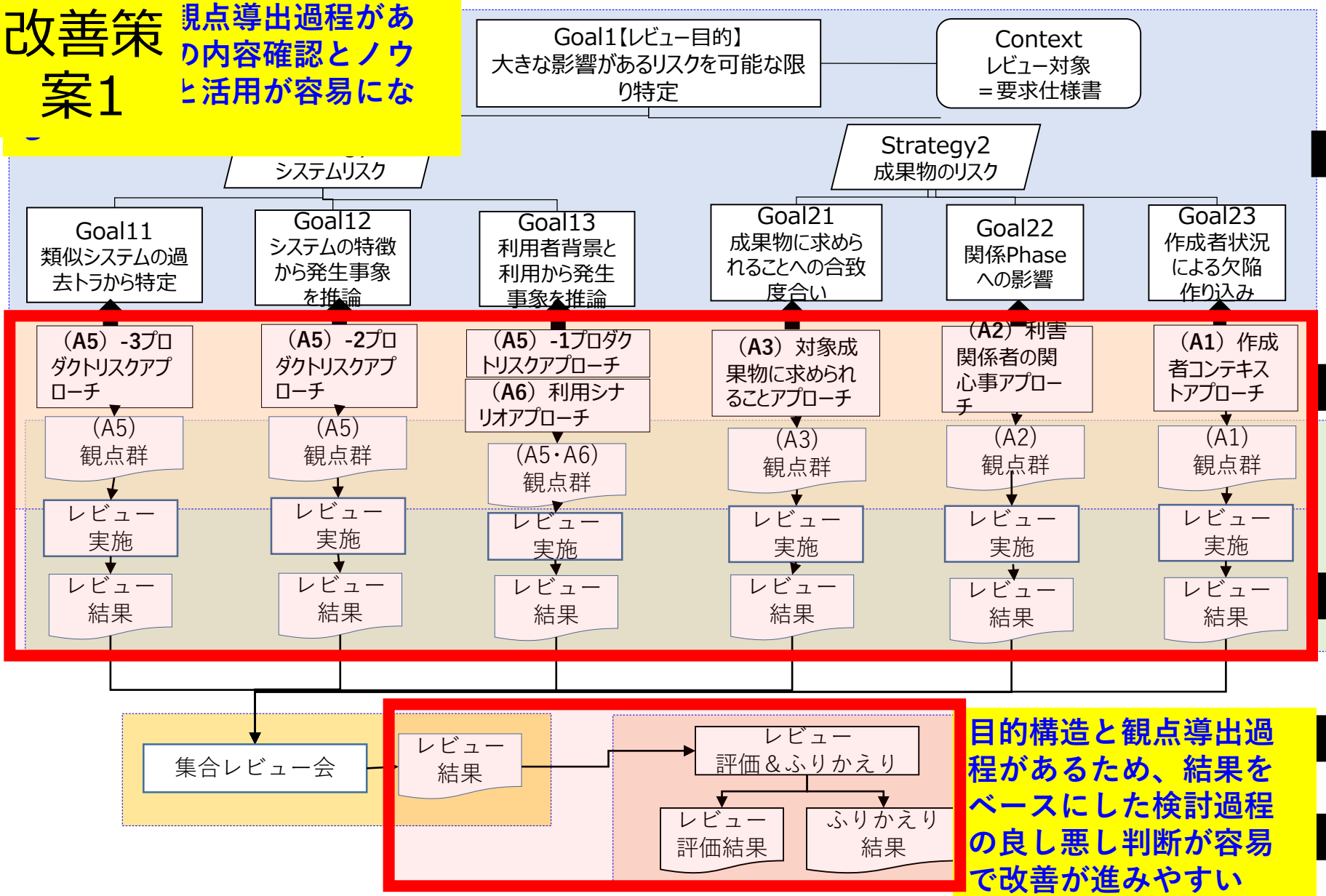
Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

さらなる具体化＝ソリューション検討

改善策案1

観点導出過程があの内容確認とノウと活用が容易にな

ワーク実施領域



目的構造と観点導出過程があるため、結果をベースにした検討過程の良し悪し判断が容易で改善が進みやすい

レビュー目的を基軸とした目的構造化表記の採用[拡張]

レビュー観点導出技法としての部品化と目的構造化表記への割り付けによるレビュー観点導出[技法バリエーション充実]

レビュー観点に基づくレビュー実施

集合レビュー会での建設的議論による結果共有

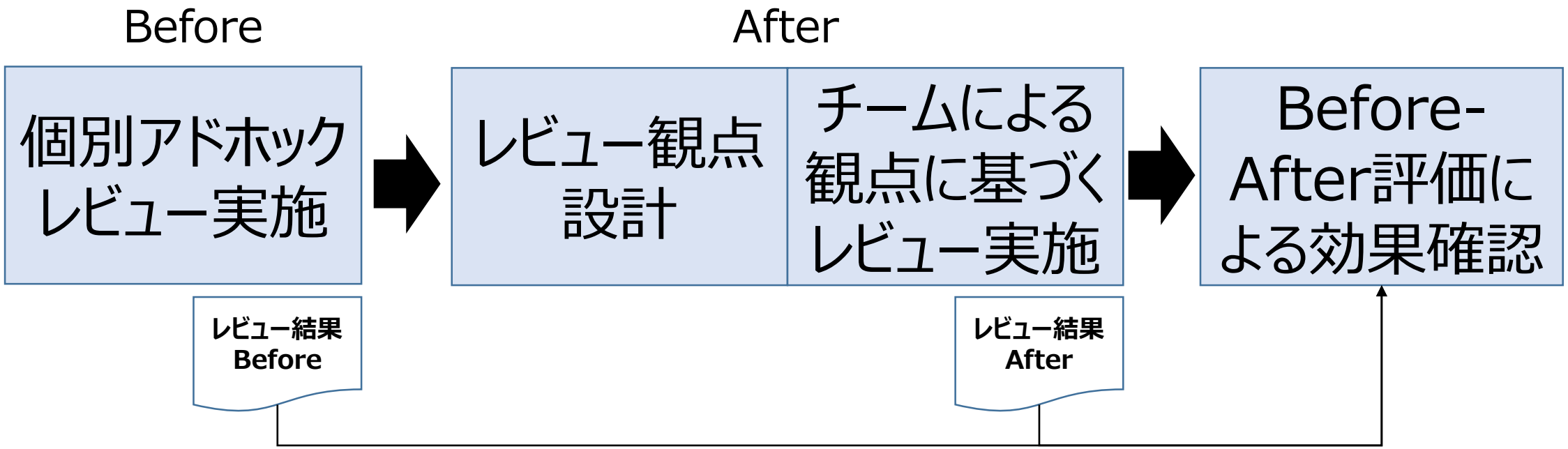
レビュー結果評価[拡張]とふりかえりによるレビュー改善検討

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

さらなる具体化＝ソリューション検討

改善策案1

アプローチの効果を実感するワークデザイン



Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

さらなる具体化＝ソリューション検討

改善策
案1

Zoom＋miroオンラインワーク実践環境構築

オンライン環境への接続が可能であれば、実施者・受講者のロケーションに左右されず運営・実施が可能

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

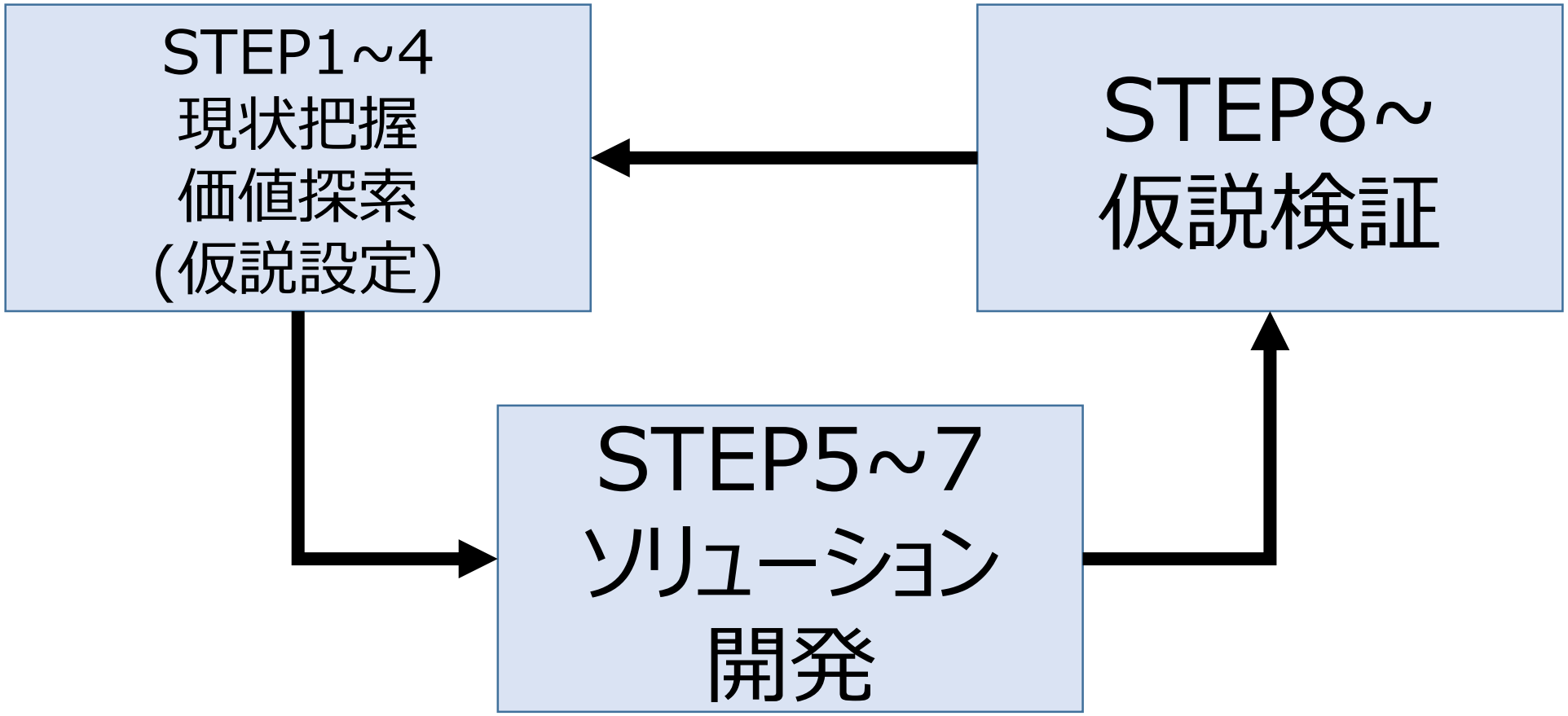
STEP6以降は実施実績にてお知らせ

- STEP6：改善目標の検討・決定
- STEP7：改善計画立案
- STEP8：トライアルによるブラッシュアップ
  →本格展開へ

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

アプローチの全体像

[仮説→実装→検証]×n　を回しながら答え(本質)に迫る



Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

■ソリューションのトライアルと評価／フィードバック獲得

改善策
案1

試行ワーク受講者による評価

	理解度	実務有効性	受講満足度
試行ワーク受講者23名評価平均	72.2/100	88.9/100	95.3/100

市場で最もニーズが高い顧客層に試行し、
ソリューションの効果や価値を評価してもらう
そして貴重なフィードバックをいただく

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

試行結果によるソリューション/アプローチの見直し

改善策案1

事象：理解度が**72.2**ポイント
事実：受講者が腕っこきQAばかり　→　仮説：納得しないと気が済まない傾向有
対応：対処しすぎる必要なし→**ワーク前の技法解説を丁寧めに行う**

ポジティブコメント例

- レビュー観点導出のレビューをしたくなる
- 観点が充分出されていると成果物への自信に繋がる
- 体系的なレビュースキルの蓄積、向上につなげられそう
- 観点設定レビューから新たなリスクが見つけられてよかった
- 観点によって発見できる問題がぜんぜん違う
- 楽しかったから、もっと食べたい！
- 自分の担当箇所にフォーカスする。これにより楽になる一方、自分の担当範囲に対する緊張感と、担当範囲外への不安が生まれる

ネガティブコメント例

- 観点導出を関係者で一緒にやらないと、チェックリスト化になってしまう
- 目的、実現手段、確認事項、の階層化がうまくできる気がしない
- 導出アプローチを正として進めてしまうと導出アプローチの漏れがあると指摘が漏れる
- レビューのための時間確保できるか
- 実践するには時間と忍耐がかかりそうです。
- レビュー導出アプローチに慣れが必要でここにもスキルレベルの差が観点に出てしまうのではないかと思いました。

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

改善策
案1

Z社ワークショップ実施実績と受講者評価結果

4点満点評価を100点満点に変換して表示

回	開催日	受講者数	理解度	有効性	満足度
8	2023/2/14	7	**90.0**	**90.0**	87.5
7	2023/1/17	11	**95.5**	**95.5**	**96.0**
6	2022/12/20	9	**91.7**	**91.7**	**90.3**
5	2022/11/29	7	**96.4**	**96.4**	**95.5**
4	2022/11/15	11	84.1	86.4	87.5
3	2022/10/25	7	**92.9**	**96.4**	**94.6**
2	2022/9/16	21	**90.0**	88.8	88.1
1	2022/7/12	15	**90.0**	89.2	89.2
		計　88名	Avg **91.3**	Avg **91.8**	Avg **91.1**

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

改善策
案1

ワーク受講者による評価

	理解度	実務有効性	受講満足度
初度試行 受講者23名 評価平均	72.2/100	88.9/100	95.3/100
Z社受講者 14回201名 評価平均	**88.1/100**	**90.5/100**	**90.2/100**

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

改善策案1

フークの成果：指摘事項の効果レベル分布

効果大・中側に厚みを持ちつつバランスよく指摘できている

収集できた全16チームの典型的な結果例

	メンバー1	メンバー2	メンバー3	メンバー4
効果大の指摘		レイアウト、画面遷移、画面機能という三つの観点は分けて説明していないため、テスト/保守は実施しづらい。属人化は初初段階でシステムに根付いてしまう可能性がある。 「承認」「経理対応」は要求の一環であるが、「一次承認者」に対する定義が明確でない。経理の処理フロー、画面、一般申請費用細画面とのデータ連携などが明確でない。	通常の業務フロー部分だが、承認されなかった場合のフローが記載なし 社内ポータルサイトからのシングルサインオンとなっているが、シングルサインオンの仕組み（仕様）の定義がない。	申請が否認となった場合のフローがない 申請が否認となった場合のステータスは何か、他にもステータスはあるのか
効果中の指摘	利用ケースごとにエラーとなるパターンの記載が無い 画面遷移図が用意されていない。	NGワード/曖昧語があり、認識齟齬が発生しうる。 「新規作成」の機能について説明されているが、画面上該当のボタンがない		申請が否認となった場合に申請者にメールは届かないのか（資料では、申請者にメールが届くタイミングは承認完了時のみ）
効果小の指摘		仕様書には目次がない、全体の構成が見えない	明細画面で「申請」「承認」した後に一覧画面に戻るのか？そのまま？	

指摘数 Avg/Team	全体に占める割合
5.3	41%
5.1	39%
2.6	20%

指摘事項の例

- 否認・再申請の手続きがない
- 性能等非機能要件が定義されていない
- 検索・データの保持期限は？
- 承認者本人申請時は誰が承認する？
- 交通費画面詳細部に誤記
- 入金待ではなく支払待では？

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

改善策
案1

※実施内容に対するコメント

受講後コメント例1

想定通りの反応～全200件コメントのうち70件(35%)

□最初に全体の説明、その後実習で実践の形で設計されており、実践する前に今からやる方法のメリットや違いがわかりやすく定着しやすかった。

□レビュー観点を事前に出して頂くのは有効。観点を普段使いするためには慣れが必要。

□レビューアのための研修でしたが、レビューイとして気付くことも多く、よかったです。

□レビューする側の視点を知ることで、成果物を作成する際の観点も磨かれていくと思う。

□改めてレビュー観点設計の大切さが身に沁みました。

□レビューの観点、全部を短い時間で浅く実践するのではなく、 一つに絞って実践することで、適切に学びを得ることができたと思います。

□事前の課題でレビューしたことを講義を聞いてから再度レビューをすることで、感じ方が変わったことが体験できとても有意義でした。

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

■アプローチの良し悪し～運用モニタリング

改善策
案1

受講後コメント

ワーク時間への反応～全73件コメントのうち8件(12%)

□時間にもう少し余裕があると良いなと思った。（4件）

□各アプローチごとの説明もう少し聞きたかった。

□短時間の研修のため、少し駆け足な感じなのが残念。

□３時間の内容としては、少し難しく。ワークの時間も足りない印象でした。

※運営環境に対するコメント

オンライン受講環境への反応～全73件のうち28件(38%)

□**miroは作業の共有がしやすく良かったです。（2件）**

□miroが使いにくい。（8件）

□miroの使用が初めてで操作方法に戸惑う事があった。（8件）

□miroに接続できずに終わった。（2件）

□受講者や講師の声が聴きとれないことがあった。（4件）

□PCが落ちて立ち上げ直しました。（2件）

□zoomが落ちて立ち上げ直しました。（2件）

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

■サービス提供と効果

改善策
案1

“JaSST”：ソフトウェアテストシンポジウム
“SS”：ソフトウェア・シンポジウム
“SQiP”：ソフトウェア品質シンポジウム

提供したソリューション実績と今後予定

2019年度	2020年度	2021年度	2022年度	2023年度

		これまでの実績		2024年度開催予定
		実施回数	受講者数	
Z社 レビュー基本ワーク	2019.7	42回	865名	6回
試行・営業 **JaSST’22東京初度試行ワーク**	2022.3			
営業・FB獲得 SS’22論文発表(企画・計画段階)	2022.6			
Solution提供 **Z社レビュー観点設計ワーク**	2022.7	**14回**	**201名**	**6回**
営業・FB獲得 **JaSST’22Review観点設計ワーク**	2022.10			
営業・FB獲得 SS’23論文発表(実践評価結果考察)	2023.6			
新Solution企画・提供 **Z社レビューパフォーマンス改善ワーク**	2023.9	**3回**	**60名**	**4回**
Solution提供 **日本テクノセンターレビュー観点設計ワーク**	2023.9			
営業・FB獲得 **SQiP’24併設レビュー観点設計チュートリアル**	2024.9			

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

SaPID適用事例

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPIDの主な事例発表

組織・チーム・個人による活用

□個人パフォーマンス改善：SaPID Bootcampにて参加者がそれぞれ実践

□チームパフォーマンス改善：SPI Japan2013＋SPI Japan2015事例発表

□組織的生産性向上施策展開：SPI Japan2012事例発表

□組織中期事業計画立案：SPI Japan2017事例発表

システム開発への活用

□未来予想(FRIモデル)に基づくプロジェクト運営：SS2020＋SPI Japan2020事例発表

□ソフトウェアレビュー改善：JaSST2016東京事例発表　他

□ソフトウェアテストプロセス改善：JaSST2018東京事例発表

□ロスコンプロジェクト分析：SPI Japan2019事例発表中の一部項目として紹介

□ODC→SaPID連携：2024/2/13 ODC分析×SaPIDオープンセミナー

DXへの活用

□SaPID→RDRA連携：JaSST2021東京チュートリアル（資料）

□現状分析→未来価値設計～コトづくり：JaSST2023東京ワークショップ

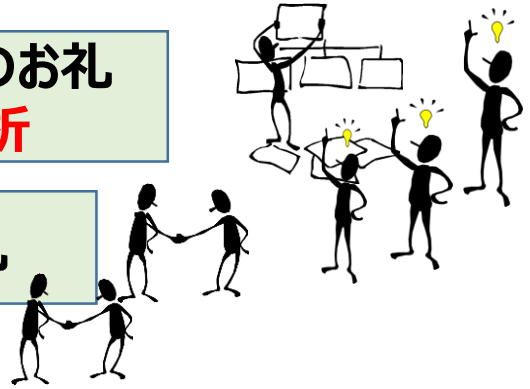
Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

チームパフォーマンス改善事例

問題の早期発見・解決＋リスク対応→手戻り減少＆リードタイム短縮

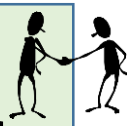
個別/チーム改善・リスク対策実践

- チーム内で発生する問題数の減少、および問題の高度化
  - 問題発生＆解決報告＆協力メンバーへのお礼＋自己タスク分析結果と自己改善宣言／チーム改善提案＋リスク要因・手戻りなどのムダ事項と対策提案　など
  - 問題発生＆解決報告＆協力メンバーへのお礼＋自己タスク分析結果／チームタスク分析



問題発見・解決中心

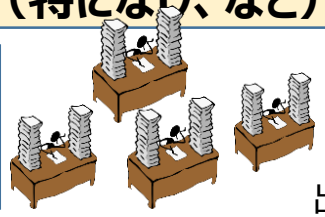
- チーム内問題発見～解決リードタイム短縮
  - 問題発生＆解決報告＆協力メンバーへのお礼
  - 問題記述・問題解決報告＋協力してくれたメンバーへのお礼



その日の問題 その日のうちに！ ふりかえり導入

- ぼんやりした境遇説明や感想（ある意味小学生的なw）／他人事のように感じる外的責任への言及／未記入（特になし、など）

個別対応者の集まり
ふりかえりなし
パフォーマンス↓



記載内容の段階的高度化＝モノゴトの見方・思考の変化

- 問題解決型→リスク・先読み型へ
- 解決手段提示→問題事象中心へ
- 個人タスク型→チーム問題発見型
- 感情面偏重→エンジニアリング的内容へ
- ぼんやりした概要記述→具体的詳細記述

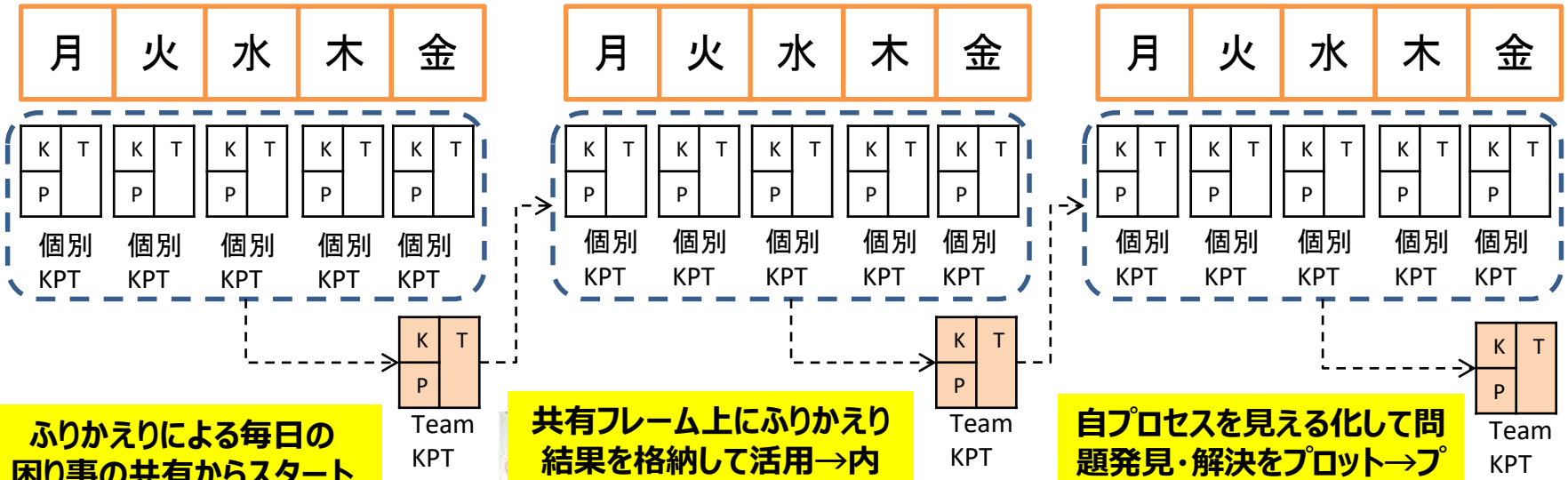
出典：SPI Japan2015　自律型プロジェクトチームへの変革アプローチ事例

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

その日の問題その日のうちに

出典：SPI Japan2015　自律型プロジェクトチームへの変革アプローチ事例

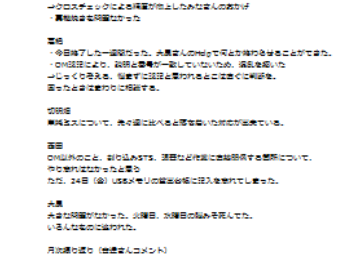
日次・週次・月次ふりかえり実践→問題発見・解決を徐々に高度化→パフォーマンス改善実践へ

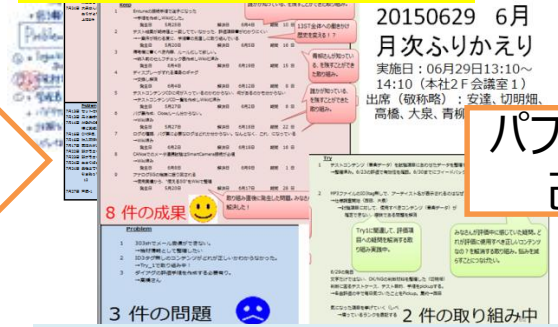


ふりかえりによる毎日の困り事の共有からスタート

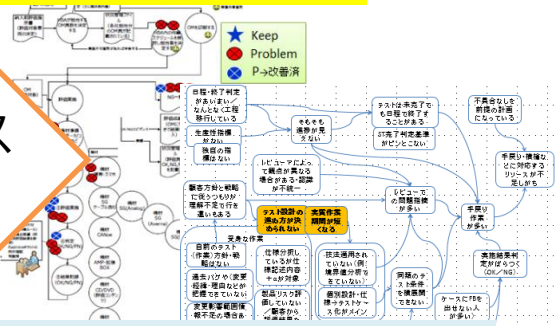
共有フレーム上にふりかえり結果を格納して活用→内容をメトリクスで管理

自プロセスを見える化して問題発見・解決をプロット→プロセス見直し実践





パフォーマンス改善へ



事前観察→暗黙の価値観を推定

チームの暗黙の価値観を見抜き問題提起→目標共有
チーム問題発見・解決の実践

達成目標を段階的に上げ、
チームパフォーマンスを上げる

必要なツールを使いこなせる
自律運営チームに仕上げる

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

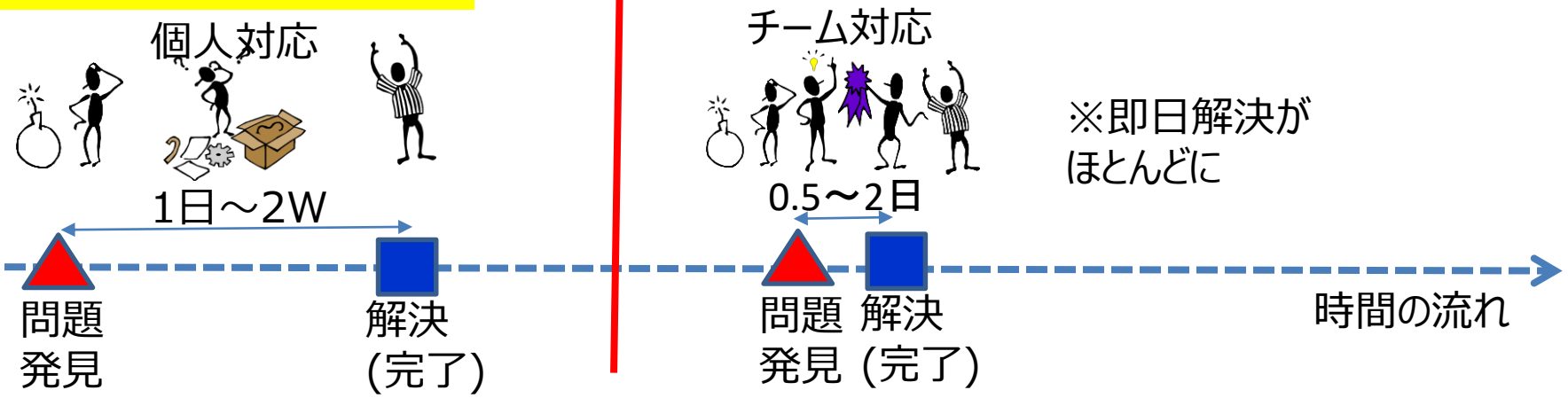
適用後のパフォーマンスの変化例

出典：SPI Japan2015　自律型プロジェクトチームへの変革アプローチ事例
http://www.jaspic.org/event/2015/SPIJapan/session3C/3C-3_ID012.pdf

■問題発生頻度＋解決工数

適用開始

■問題発見－解決リードタイム



※即日解決がほとんどに

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

チームパフォーマンス改善試行の観察結果

比較事項	Before	After
ミーティング	ほとんどなし。自作業に専念できる。	ミーティングに時間が割かれるようになった。緊急時や作業立て込み時は時間のやりくりが難しくなった。
大事にすること	各自が役割を果たせばよい。	個人ではなく“チームとしての効率の良さ”として捉える。
ノウハウ	属人的。個人にノウハウが集中。 初めて実施する作業の立ち上がりが遅く、実務・管理両面で苦労していた。	チームとして共有。
問題発見・解決	個人対応で場当たり的。 他者の状況が見えず対応が後手後手。	どのような問題が起こっているのかが、全体で把握できる。 問題発生したらメンバーで解決する。 早めに解決策を取れることが増えた。
改善	指示がなければ特段実施しない。	これまで課題だった再発防止ができた。 日々必要な改善を実践している。 問題発見・解決と分けて実践していない。同じ困りごとが発生しそう→改善を実践
生産性など	新規参入者や初着手作業でノウハウが不明でミスが多く、生産性も悪い。	スピードが上がった。最終チェックNGが減少。新規参入者が短時間でミス低減ができ生産性の向上が計れる。

出典：SPI Japan2015　自律型プロジェクトチームへの変革アプローチ事例
http://www.jaspic.org/event/2015/SPIJapan/session3C/3C-3_ID012.pdf

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

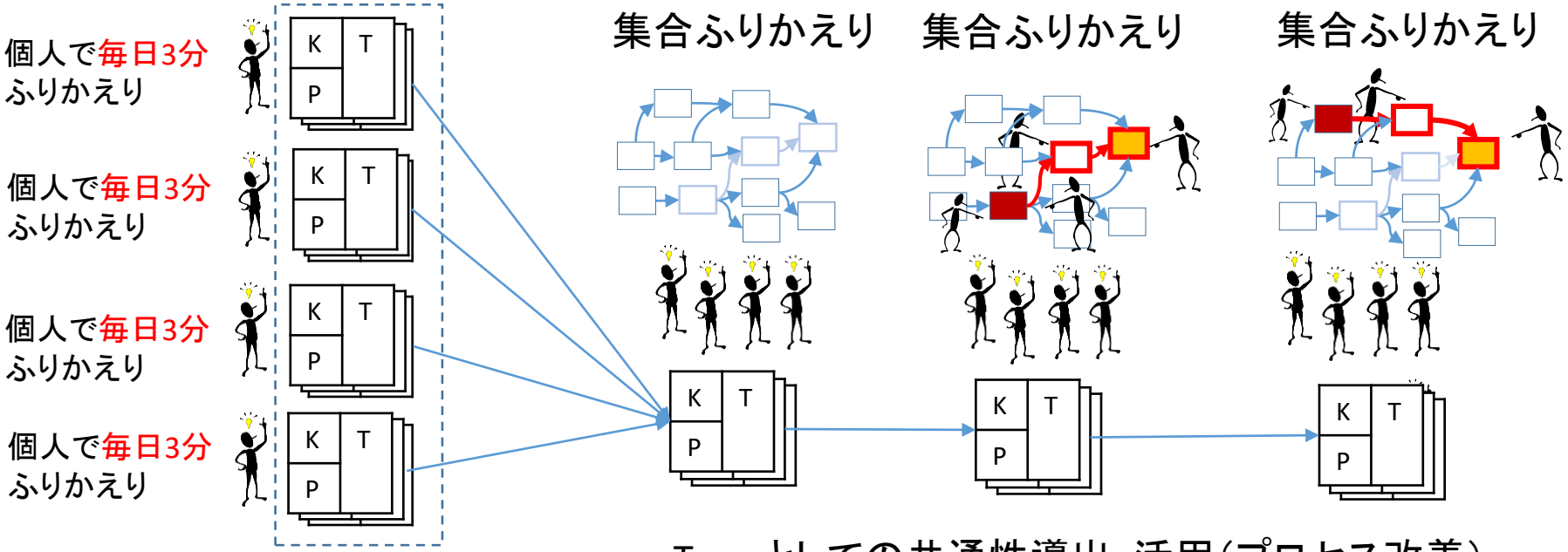
過去の体験・経験から学ぶ
個別改善→パフォーマンス改善への発展

日次個別ふりかえり
とチーム共有・活用

週次Team
集合ふりかえり

月次・Phase
集合ふりかえり

四半期・Final
集合ふりかえり



Teamとして毎日共有し、
主に日々の問題解決・
個別改善に活用

Teamとしての共通性導出・活用（プロセス改善）
ふりかえり結果情報のサマリ→構造化分析による全体俯瞰
ボトルネックの解消／最も効果的な改善ポイントの特定による
パフォーマンス改善の実践

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

SaPIDの主な事例発表

組織・チーム・個人による活用

□個人パフォーマンス改善：SaPID Bootcampにて参加者がそれぞれ実践

□チームパフォーマンス改善：SPI Japan2013＋SPI Japan2015事例発表

□組織的生産性向上施策展開：SPI Japan2012事例発表

□組織中期事業計画立案：SPI Japan2017事例発表

システム開発への活用

□未来予想(FRIモデル)に基づくプロジェクト運営：SS2020＋SPI Japan2020事例発表

□ソフトウェアレビュー改善：JaSST2016東京事例発表　他

□ソフトウェアテストプロセス改善：JaSST2018東京事例発表

□ロスコンプロジェクト分析：SPI Japan2019事例発表中の一部項目として紹介

□ODC→SaPID連携：2024/2/13 ODC分析×SaPIDオープンセミナー

DXへの活用

□SaPID→RDRA連携：JaSST2021東京チュートリアル（資料）

□現状分析→未来価値設計～コトづくり：JaSST2023東京ワークショップ

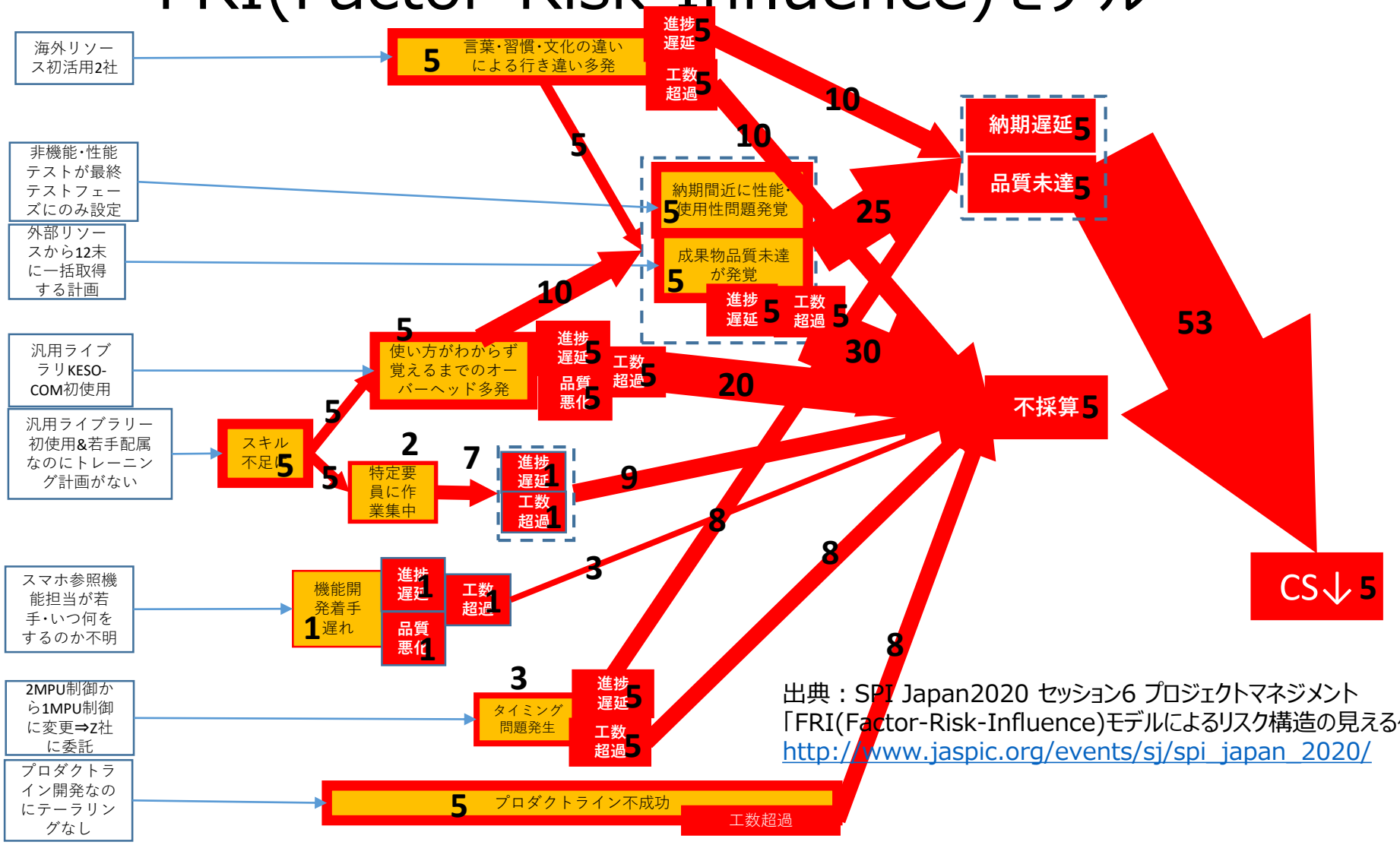
Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

よくあるリスク・問題列挙型管理表（例）

優先度	着目した内容とリスク・影響	発生可能性	影響度	リスク値	対処内容	完了日
1	取得計画ではS社に要求分析～統合テストまでを請負委託することになっているが、その成果物は12末に一括取得となっている。 S社はブリッジSE配置で、プロパーSEが割り当てされておらず体制的にも脆弱。 →慣習・商習慣、認識違いにより意図しない状態で納入され、フェーズ期日遅延、手戻り工数増、品質要件実装不備などに発展する可能性がある。	5	5	25	S社担当プロパーSEを設置。 プロジェクトとして必要なものを必要なタイミングで取得できる契約内容となっているかを事前に確認し、先方と合意する。 さらに定期的に作業成果物レビューを実施できる契約とする。	10/2
2	生産性改善施策として、汎用ライブラリ（KESO-COM）を使用する”とあるが、受入計画ではB社から201X/7/15に一式受取りとなっている。特にトレーニング計画も立案されていない。 →慣れないライブラリの使い方がわからず逆に非効率に陥る可能性がある。	5	5	25	汎用ライブラリー担当リーダを設置。 事前オリエンテーション、トレーニングを計画し、実施する。	10/12
3	取得計画では、S社・X社・Y社・A社　納期12/31に一式を受け取る計画である。 →一気に4社の成果物一式をフェーズの終わりに受け取ることで、要求品質未達成による受取時トラブルや納期遅延がそのままプロジェクト遅延につながる可能性がある。	3	5	15	複数フェーズ委託の場合、最後にすべてを受け取るのではなく、フェーズ単位で受け取る計画とする。 複数機能委託の場合は、機能単位で受け取る計画とする。 定期的に作業成果物レビューを実施できる契約とする。	9/30
4	テスト計画では、ソフトウェア適格性確認テストに非機能要件テスト、性能テストが含まれている。それ以前のテストには含まれていない。 →終盤のテストで性能や使用性問題が発覚すると納期遅延や品質悪化、標準資産化断念などにつながる可能性がある。	3	5	15	非機能要件テスト、性能テストのうち、最終テストフェーズ以前に実施可能な内容を前倒しする。	9/23

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

プロジェクトリスク・問題表現形式（例）
FRI(Factor-Risk-Influence)モデル



出典：SPI Japan2020 セッション6 プロジェクトマネジメント
「FRI(Factor-Risk-Influence)モデルによるリスク構造の見える化」
http://www.jaspic.org/events/sj/spi_japan_2020/

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPIDの主な事例発表

組織・チーム・個人による活用

□個人パフォーマンス改善：SaPID Bootcampにて参加者がそれぞれ実践

□チームパフォーマンス改善：SPI Japan2013＋SPI Japan2015事例発表

□組織的生産性向上施策展開：SPI Japan2012事例発表

□組織中期事業計画立案：SPI Japan2017事例発表

システム開発への活用

□未来予想(FRIモデル)に基づくプロジェクト運営：SS2020＋SPI Japan2020事例発表

□ソフトウェアレビュー改善：JaSST2016東京事例発表　他

□ソフトウェアテストプロセス改善：JaSST2018東京事例発表

□ロスコンプロジェクト分析：SPI Japan2019事例発表中の一部項目として紹介

□ODC→SaPID連携：2024/2/13 ODC分析×SaPIDオープンセミナー

DXへの活用

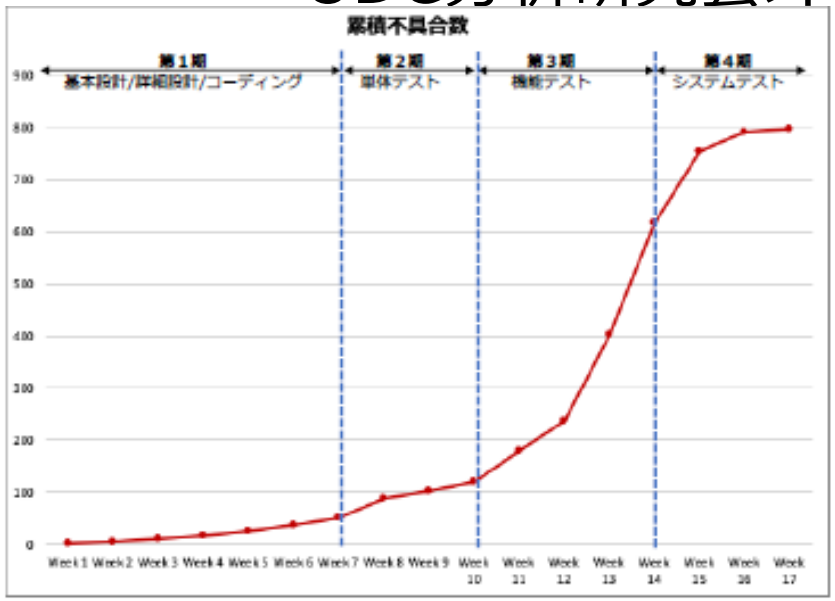
□SaPID→RDRA連携：JaSST2021東京チュートリアル（資料）

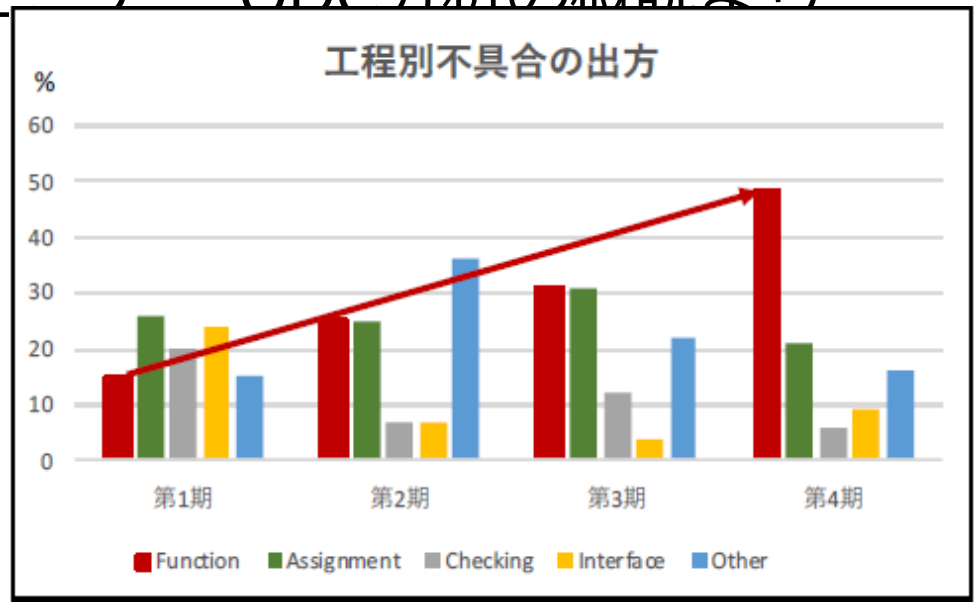
□現状分析→未来価値設計～コトづくり：JaSST2023東京ワークショップ

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

ODC分析適用事例1

ODC分析研究会オープンセミナー ODC分析の概説より





適用された開発プロセスに沿って計画どおり開発・テストを進めて、以下のような判定で「出荷可能」と判断した。

1) 各工程で定義された開発・テスト活動は、すべて完了した。
2) 摘出された不具合は、すべて修正した。
3) 摘出された不具合検出数は、おおむね品質計画どおりの傾向を示し、
4) システム終盤に来て、新しく摘出される不具合数は収束している。

指摘1：Function(機能性)に関わる不具合の前工程からの漏れ
指摘2：残存不具合(Function) 対応によるシステムテストへの影響

示唆される対応策1：設計工程(設計レビューやコード検証)での機能性に対する次のレビュー観点の見直しが必要である。

- 要求仕様の詳細化の十分性
- 要求仕様と設計仕様の整合性
- コード・インスペクションのレビューアの経験値

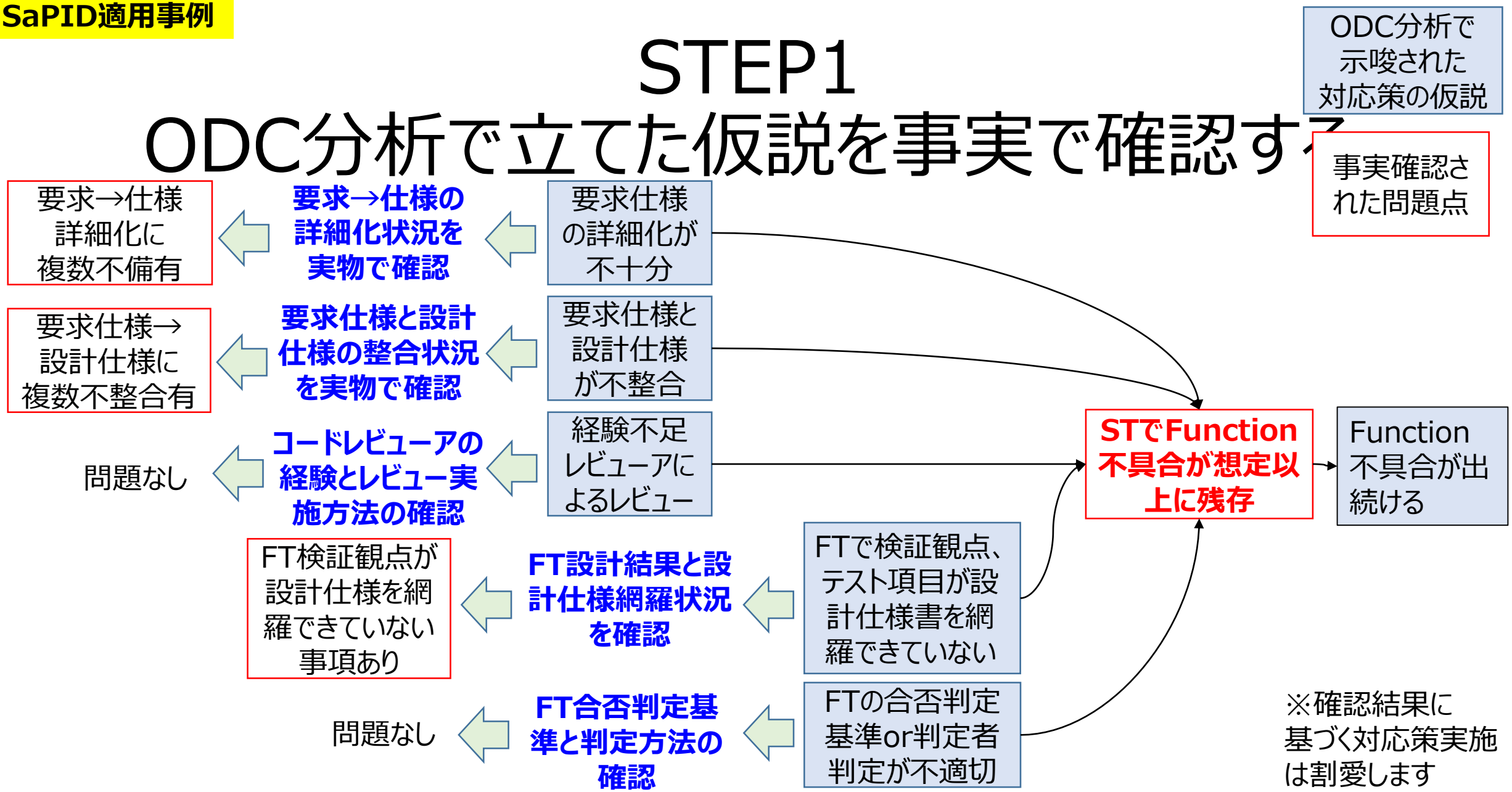
示唆される対応策2：機能テストのカバレージの網羅性と詳細性および実施体制の検証が必要である。以下の点を検証する。

- 機能テスト(統合テスト)での、検証観点、テスト項目が設計仕様書を網羅しているか。（ここで食い止めるべき）
- テスターの選定は適正か、テストの合否判定基準は正しく文書化され教育されているか。

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPID適用事例

STEP1
ODC分析で立てた仮説を事実で確認する



Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPID適用事例

STEP2-1
フェーズ内作り込み要因の状態を確認する

事実確認済
問題点

新たに事実
確認された
問題点

現状分析結果がどこにも存在しない

変更要求の仕様化に一部失敗していた

要求なのか仕様なのかが不確かな記述形態になっている

要求→仕様の詳細化で不備が出た要因は？

要求→仕様
詳細化に
複数不備有

複数の要求をそのまま設計仕様化し、詳細化不足になっている

設計仕様のみ記述する形式となっている(どの要求に対する設計仕様なのかは要求仕様書と突合しないとわからない)

要求仕様→設計仕様の複数不整合が出た要因は？

要求仕様→
設計仕様に
複数不整合有

テスト設計後の仕様変更への対応がコード修正のみとなっていた

基本的に*CPM法*でテスト設計を行っていたが、テスト設計後の仕様変更発生箇所については未対応であった

FT検証観点が設計仕様を網羅できていない要因は？

FT検証観点が
設計仕様を網
羅できていない
事項あり

※確認結果に
基づく対応策実施
は割愛します

※CPM法：Copy and Paste and Modify法（電気通信大にしさんが命名）

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPID適用事例

STEP2-2
フェーズ外流出要因を確認する

事実確認済
問題点

新たに事実
確認された
問題点

要求→仕様
詳細化に
複数不備有

**要求→仕様の詳細化
不備をレビューで未検
出？その要因は？**

レビュー対象の記述内
容に反応するだけのア
ドホックレビューで見逃
されていた

要求仕様→
設計仕様に
複数不整合有

**要求仕様→設計仕様
の複数不整合をレ
ビューで未検出？
その要因は？**

レビュー対象の記述内
容に反応するだけのア
ドホックレビューで見逃
されていた

**STでFunction
不具合が想定以
上に残存**

Function
不具合が出
続ける

FT検証観点が
設計仕様を網
羅できていない
事項あり

**FT検証観点が設
計仕様を網羅で
きていないのをレ
ビューで未検出？
その要因は？**

FT設計結果レ
ビューが未実施
であった

※確認結果に
基づく対応策実施
は割愛します

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPID適用事例

STEP3
認した事実情報を構造化する

作り込み要因

流出要因

要求定義

現状分析結果がどこにも存在しない

要求なのか仕様なのかが不確かな記述形態

変更要求を処理する際に仕様への展開に一部失敗していた

レビュー対象の記述内容に反応するだけのアドホックレビューで見逃されていた

基本設計

複数の要求をそのまま設計仕様化し、詳細化不足になっている

設計仕様のみ記述する形式となっている
(どの要求に対する設計仕様なのかは要求仕様書と突合しないとわからない)

レビュー対象の記述内容に反応するだけのアドホックレビューで見逃されていた

機能テスト

テスト設計後の仕様変更への対応がコード修正のみとなっていた

基本的に*CPM法*でテスト設計を行っていたが、テスト設計後の仕様変更発生箇所については未対応であった

FT検証観点が設計仕様を網羅できていない事項あり

FT設計結果レビューが未実施であった

STでFunction不具合が想定以上に残存

Function不具合が出続ける

※確認結果に基づく対応策実施は割愛します

※CPM法：Copy and Paste and Modify法（電気通信大にしさんが命名）

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPID適用事例

STEP4
再発防止ポイントを特定する

作り込み要因

流出要因

これらすべてを実施するのかについては改善施策検討結果と割り振れるリソース等の兼ね合いで決定します

要求定義

- 現状分析結果がどこにも存在しない
- 要求なのか仕様なのかが不確かな記述形態
- 変更要求を処理する際に仕様への展開に一部失敗していた
- レビュー対象の記述内容に反応するだけのアドホックレビューで見逃されていた

基本設計

- 複数の要求をそのまま設計仕様化し、詳細化不足になっている
- 設計仕様のみ記述する形式となっている（どの要求に対する設計仕様なのかは要求仕様書と突合しないとわからない）
- レビュー対象の記述内容に反応するだけのアドホックレビューで見逃されていた

機能テスト

- テスト設計後の仕様変更への対応がコード修正のみとなっていた
- 基本的に*CPM法*でテスト設計を行っていたが、テスト設計後の仕様変更発生箇所については未対応であった
- FT検証観点が設計仕様を網羅できていない事項あり
- FT設計結果レビューが未実施であった

STでFunction不具合が想定以上に残存 → Function不具合が出続ける

※確認結果に基づく対応策実施は割愛します

※CPM法：Copy and Paste and Modify法（電気通信大にしさんが命名）

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPID適用事例

STEP5
再発防止施策を検討する

作り込み要因

流出要因

再発防止ポイント

要求定義

現状分析結果がどこにも存在しない

要求なのか仕様なのかが不確かな記述形態

レビュー対象の記述内容に反応するだけのアドホックレビューで見逃されていた

※基本設計・機能テストの再発防止施策検討は割愛します

① 施策案1

現状分析結果を成果物として共有する

既存記述に要求、仕様を示す識別子を付与する

実践容易性：低1～中2

有効性：中2

② 施策案2

理由-要求-仕様をセットで記述するテンプレートに沿って表現する(USDM)

実践容易性：低1～中2

有効性：高3

評価結果からこちらを選択

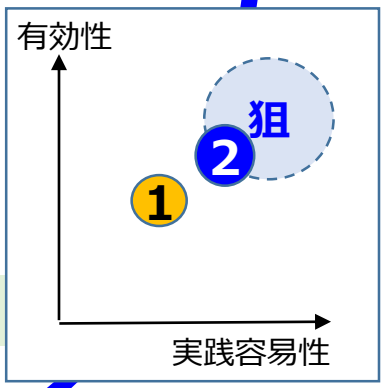
要求仕様書レビュー時の要求-仕様の十分性確認観点を必須とする

実践容易性：低1～中2

有効性：高3

実践容易性：中2～高3

有効性：高3



Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

自由形式の要求仕様記述例

■前提条件
電源ON/OFFはコンセントの抜き刺しで行われる。
電源ON時は00:00を設定し、タイマ動作停止モードとする。
00:00の前2桁は分、後ろ2桁は秒を示す。
■機能
設定した時間から1秒ずつカウントを減らし、00:00になった時点でブザーを鳴らす。
＜時間設定＞
・タイマ時間設定は“分のみ”対象とする。（秒はそのままとする）
・初期値は00分、最大値は60分
・タイマ動作停止モードでタイマボタンを3秒以上長押すると、時間表示をすべてリセット（00:00）して点滅し、時間設定モードに遷移する。
・時間設定モードでタイマボタンを1回押下（ 3秒未満）すると1分カウントアップし、ブザー音“ピ”を鳴らす。（60分表示でタイマボタンを押すと00分とする）
・時間設定モードでタイマボタンを3秒以上長押すると、時間表示点滅を終了（点灯）し、タイマ動作停止モードに遷移する。
＜タイマ動作開始・停止＞
時間設定が00:00を超える場合、タイマーボタンを1回押下（ 3秒未満）すると計測を開始する。設定時間から1秒ずつカウントを減らし、その経過をリアルタイムで時間表示する。また計測中にタイマボタンを1回押下（ 3秒未満）するとタイマ動
＜タイムアッ
カウントダウ 00になった時点でブザーを鳴らす。ブザー音は“ピピピピ、ピピピピ、 イマボタンを1回押下（ 3秒未満）するとブザー音を止める。

参考：[入門+実践]要求を仕様化する技術・表現する技術 -仕様が書けていますか?

USDM形式の要求仕様記述例

要求	A1	タイマの電源をON／OFFする	
	理由	タイマを使用するとき電源を入れたい。使わないときはOFFにしたい。	
仕様	□	A1-1	コンセントに差すと電源ON。時刻に00:00を設定・表示し、タイマ動作停止モードとなる。
	□	A1-2	コンセントを抜くと電源OFF。
要求	A2	タイマに計測する時間を設定する	
	理由	計測する時間が用途で異なるため自由に設定したい。	
仕様	□	A2-1	タイマ動作停止モードでタイマボタンを長押すると、時間表示をすべてリセット（00:00）して点滅し、時間設定モードに遷移する。
	□	A2-2	タイマボタン押下すると1分カウントアップ、ブザー音“ピ”を鳴らす。(60分表示でタイマボタン押下→00分とする/3000～4000Hz、40dB前後の音)
	□	A2-3	時間設定モードでタイマボタンを長押すると、時間表示点滅を終了（点灯）し、タイマ動作停止モードに遷移する。
要求	A3	タイマ計測を開始する／一時停止する	
	理由	利用者の都合やタイミングで計測を開始したり、一時止めたり（再開したり）したい。	
仕様	□	A3-1	時間設定が00:00を超える場合、タイマボタンを押下すると計測を開始する。1秒ずつカウントダウンし、その経過をリアルタイムで時間表示する。
	□	A3-2	計測中にタイマボタンを押下するとタイマ動作を停止する。
要求	A4	設定時間が経過したら音で知らせる／音を止める	
	理由	多少離れた場所でも設定した時間が経過したことを知りたい／把握できたら音を止めたい。	
仕様	□	A4-1	カウントダウンの結果、00:00になった時点でブザーを鳴らす。ブザー音は“ピピピピ、ピピピピ、・・・”とし、3000～4000Hz、40dB前後の音とする。

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPID適用事例

STEP6-1
再発防止施策の展開方法を検討する

要求仕様書は理由-要求-仕様をセットで記述するテンプレートに沿って表現する(USDM)

①基本タスク
USDMテンプレートを準備する。
典型的な記載事例、Q&Aを作成して共有する。
最初に着手するメンバーにUSDM説明会＆トレーニングを行う／単元ごとにビデオ化していつでも参照できるようにする。→以降は着手前にビデオを視聴してから着手する。
問合せ窓口担当者を割当て、問合せ実績をQ＆Aに反映する。
レビュー結果の指摘内容の推移をモニタリングして、定着度を把握し、施策実践期間をコントロールする。

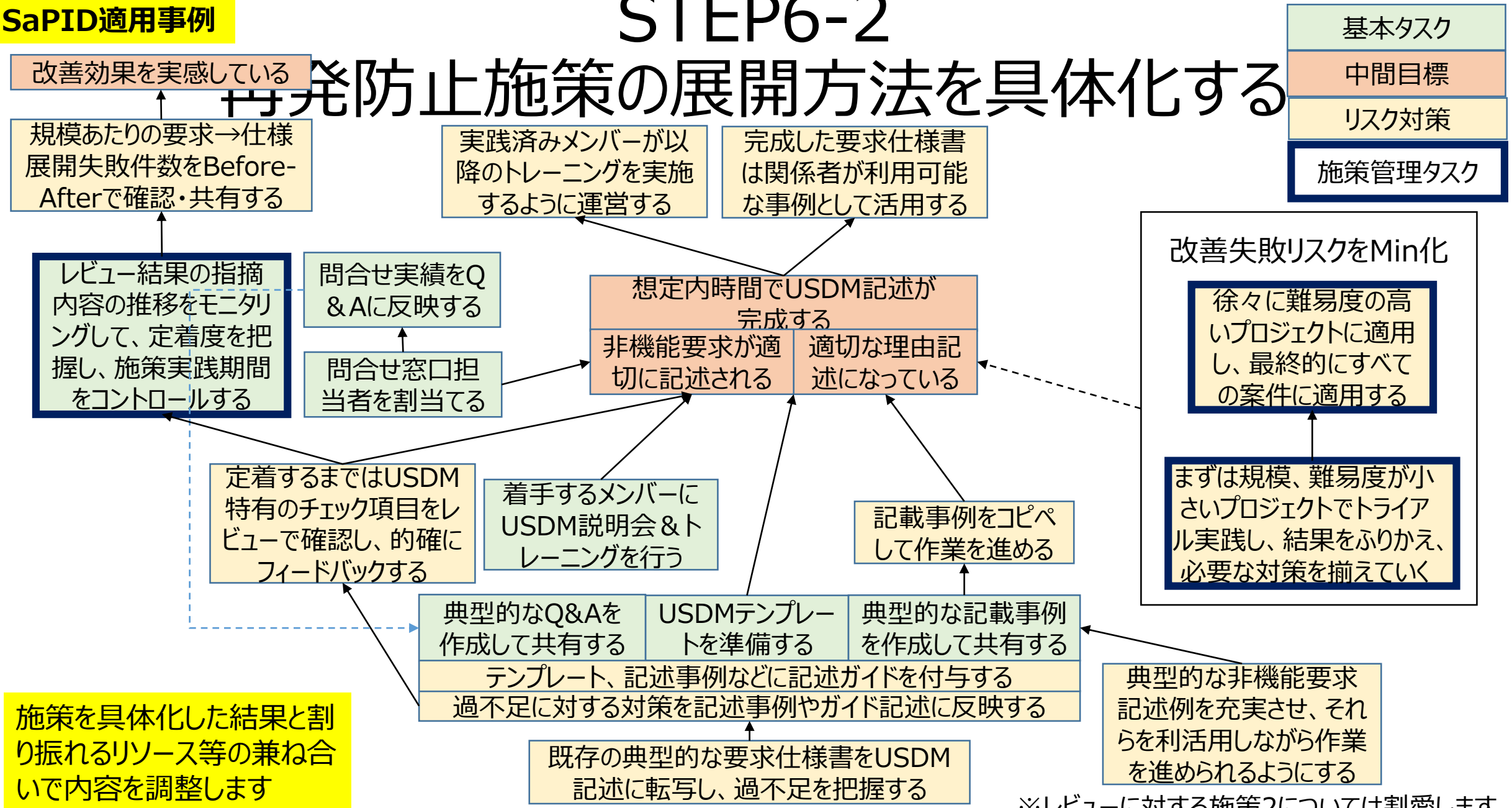
※以外の施策については割愛します

②リスク	③中間目標	④リスク対策
ゼロからUSDM記述を作成すると時間がかかりすぎる。	想定内時間でUSDM記述が完成する。	□既存の典型的な要求仕様書をUSDM記述に転写し、過不足を把握する。過不足に対する対策を明確にして記述事例やガイド記述に反映する。 □まずは規模、難易度が小さいプロジェクトでトライアル実践し、結果をふりかえ、必要な対策を揃えていく。徐々に難易度の高いプロジェクトに適用し、最終的にすべての案件に適用する。 □記載事例をコピペして作業を進める。 □定着するまではUSDM特有のチェック項目をレビューで確認し、的確にフィードバックする。 □完成した要求仕様書は関係者が利用可能な事例として活用する。 □実践済みメンバーが以降のトレーニングを実施するように運営する。 □テンプレート、記述事例などに記述ガイドを付与する。 □典型的な非機能要求記述例を充実させ、それらを利活用しながら作業を進められるようにする。 □規模あたりの要求→仕様展開失敗件数をBefore-Afterで確認・共有する
これまで記述していなかった項目を記述するため時間がかかる。		
理由欄が空欄のまま／自社都合の理由が記述される。	適切な理由記述になっている。	
非機能要求が記述されない。	非機能要求が適切に記述される。	
改善効果が実感できない	改善効果を実感している	

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPID適用事例

STEP6-2
再発防止施策の展開方法を具体化する



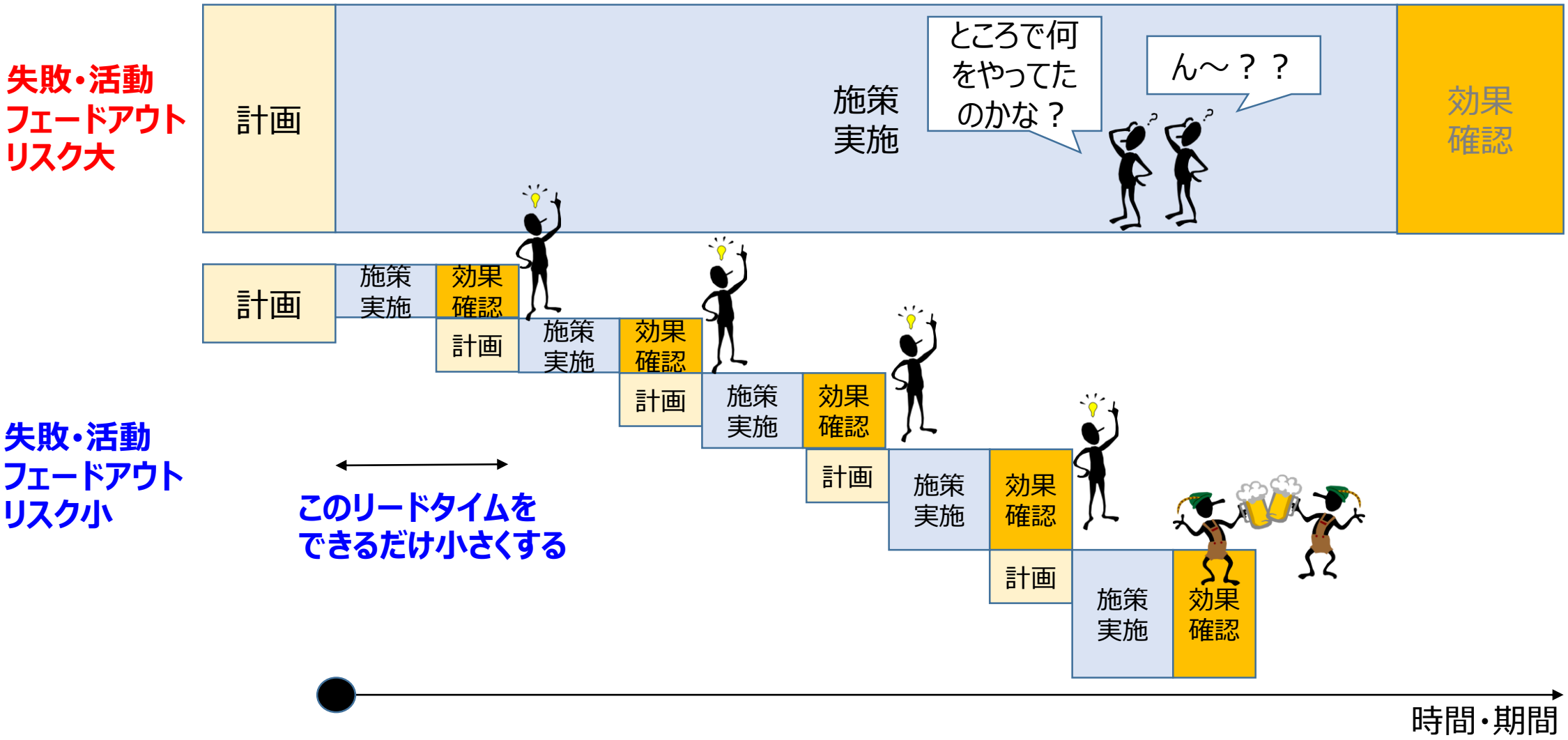
施策を具体化した結果と割り振れるリソース等の兼ね合いで内容を調整します

※レビューに対する施策2については割愛します

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

STEP7以降

パフォーマンス改善では効果実感や手ごたえを早期に



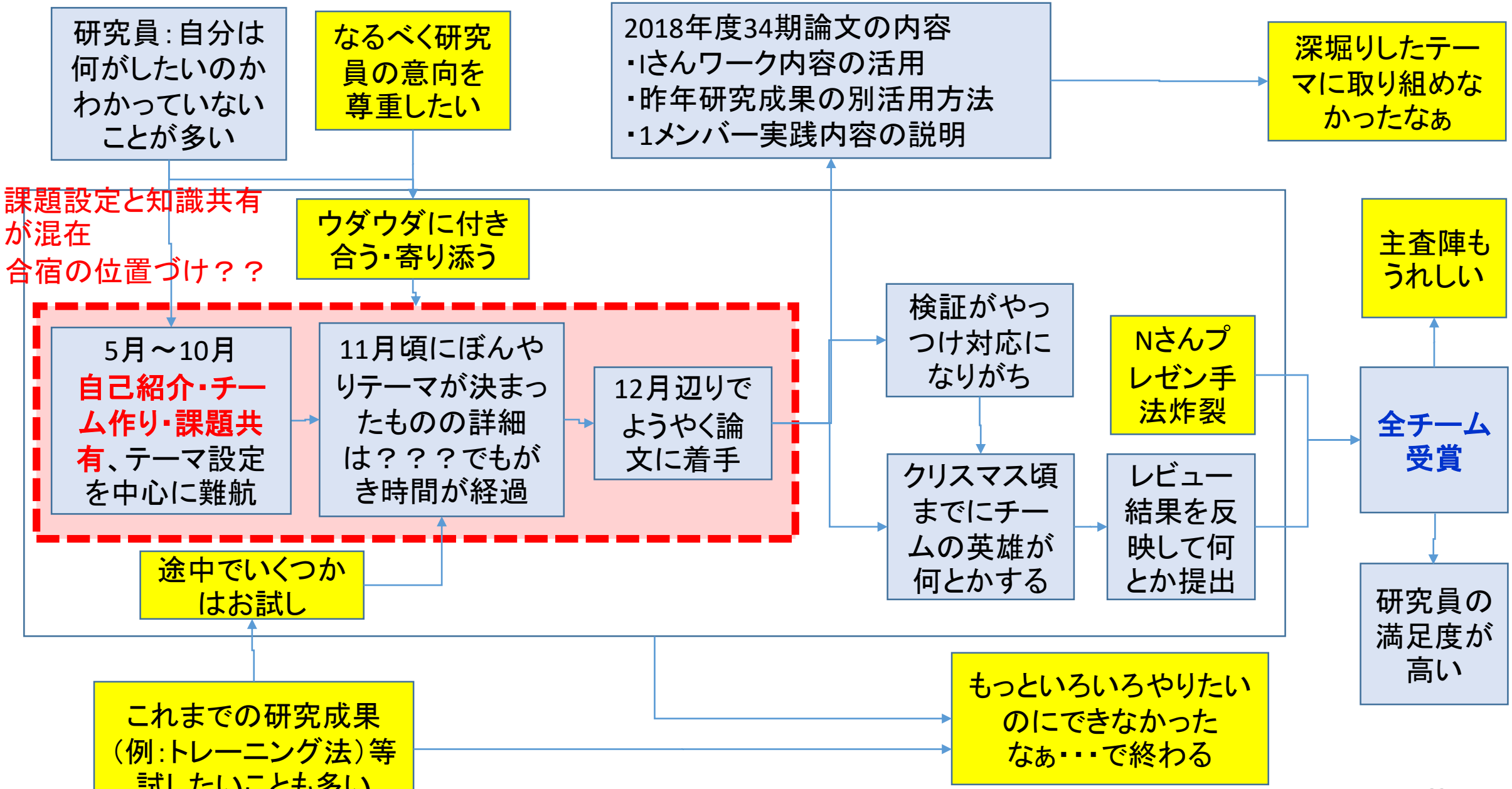
Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPIDを活用したSQiP研究会マネジメント(例)

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

2018年度実績分析

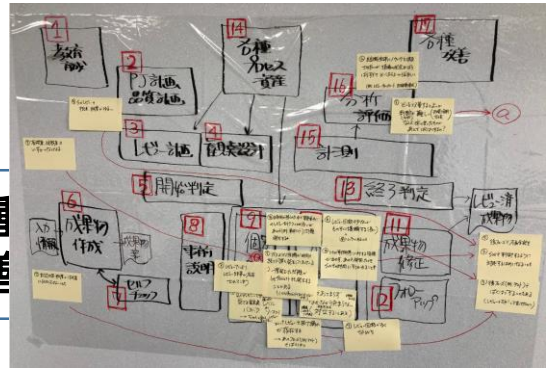
S研究会レビュー分科会の2018年度運営はこうだったんじゃないか構造図



Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

2019年度5月(第1回)

レビュー領域の困りごと・問題・課題マッピング　20190510版



1 教育・育成

(5)SWレビューの技法・粒度が緩い

2 PJ計画・品質計画

3 レビュー計画

4 観点設計

14 各種プロセス資産

(2)チームにより背景が異なる

17 各種改善

(2)組織全体にノウハウを波及したいが現場の状況がばらばらすぎて困る

16 分析評価

(5)どこを改善するとよいのか判断が難しい（知見活用できていない）

a

15 計測

6 成果物作成

入力情報

成果物案

(7)記述内容・粒度が作成者に伝わっていないことがある

セルフチェック

7

(7)有識者はいずれいなくなる

5 開始判定

(8)レビューアにより結果が左右されてしまう

(2)設計レビュー結果のばらつきが大きい

(3)HW担当が欠席しがちなのでHW制御系への指摘が出せないことも

(1)レビューアが見てる観点・掘り下げレベルがバラバラ

事前説明

個別Review

集合Review

(5)設計レビューで抜け漏れがある

(7)同じような指摘がなかなかなくならない（指摘内容が以降で活用できていない）

(7)レビューの結果が人に依存している

a

(1)一度発生した問題が他プロジェクトでも発生することがある（知見共有できていない）

記録

13 終了判定

(1)解決策検討で紛糾し、時間がかかることも

(3)クイックDR時変更点に絞って実施する

(6)レビュー回数は少ないが猛烈に議論する（長い）→ムラがあること

11 成果物修正

(7)レビューの回数が多くなることがある

フォローアップ

2

完成版成果物

(5)後フェーズで不具合が出ることが

(5)テストフェーズで爆発することも

(3)テスト中に問題発覚した際に、SWで対処としわ寄せが来ることも

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

2019年度6月(第2回)

「論文概要」
「テーマ検討ワークシート」
＋本人プレゼンから読み取れる概要

	解決したい問題・課題（要因と結果）		解決手段（案）	
Hさん	いつも同じミニDR（チームレビュー）をしている	品質は良いかもしれないけどレビューの効率が悪い	フェーズ・状況に応じて手法を臨機応変に変更する	
Yさん	レビューのやり方がまちまち、バラバラ	レビュー密度低・指摘0・実施記録なしの場合もある	レビューのやり方を標準化、共有＋実践内容を可視化	レビューシナリオの事前共有
Sさん	レビューアスキルバラバラ→観点不明確 短納期→レビュー時間裂けない	レビュー結果がメンバーのスキルに左右される&レビュー実施時間が長い 安定した成果物が作成できない 修正の修正が発生する	短時間でメンバースキルに左右されずにできる方法で実施する	
Fさん	形骸化したレビューor実施方法がばらついている	重大欠陥が後工程で見つかる	適切なレビュー実践になっているかを可視化する	重大欠陥の抜け漏れがレビュー結果から把握できる
SBさん	レビューアの観点・見方が個人依存 実施時間がバラバラ	重大欠陥を見逃す人もいる	レビュー実施傾向の可視化 レビューア毎のやり方を共有する	得手不得手によりレビューア割当て レビューノウハウ活用
Kさん	変更点に着目したレビュー実施	指摘が変更部分に集中→思わぬところに存在する影響箇所を見逃す あとで抜け漏れが見つかる	レビューアの役割分担制	変更点可視化シート(追加・削除・修正) 変更点に寄らないレビュー手法の提案
KSさん	目的・論点がレビューアに分散（依存）して言いたい放題 レビュー時間膨大	レビュー効率、重大欠陥除去率が異なる 終了時に作成者の顔が曇る	検出シナリオ決定法 作成者が]目的・観点設定 →レビューア選定→検出可能な欠陥の推測]	
Hさん	仕様から読み取れていない（非機能等）事項がある	後になってアーキテクチャによるスケーラビリティ不具合が発生している	当初からの変化への対応	テストコードもレビュー？

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

2019年度6月(第2回)

レビュー領域の困りごと・問題・課題マッピング　20190614版

1 教育・育成

レビューア スキルバラバラ

2 PJ計画・品質計画

14 各種プロセス資産

17 各種改善

3 レビュー計画

16 分析評価

目的・論点がレビューアに分散(依存)

レビューアの観点・見方が個人依存

観点設計

観点不明確

15 計測

短納期

早期リリース必達 Agile開発

第三者的に品質を見る必要あり

6 成果物作成

入力情報

成果物案

セルフチェック

7

5 開始判定

13 終了判定

完成版成果物

変更点に着目したレビュー実施

8 事前説明

9

形骸化したレビューor実施方法がばらついている

いつも同じミニDR(チームレビュー)をしている

レビューア言いたい放題

個別Review

集合Review

仕様から読み取れていない(非機能等)事項がある

レビューのやり方がまちまち、バラバラ

レビュー時間膨大

レビュー時間裂けない

実施時間がバラバラ

11 成果物修正

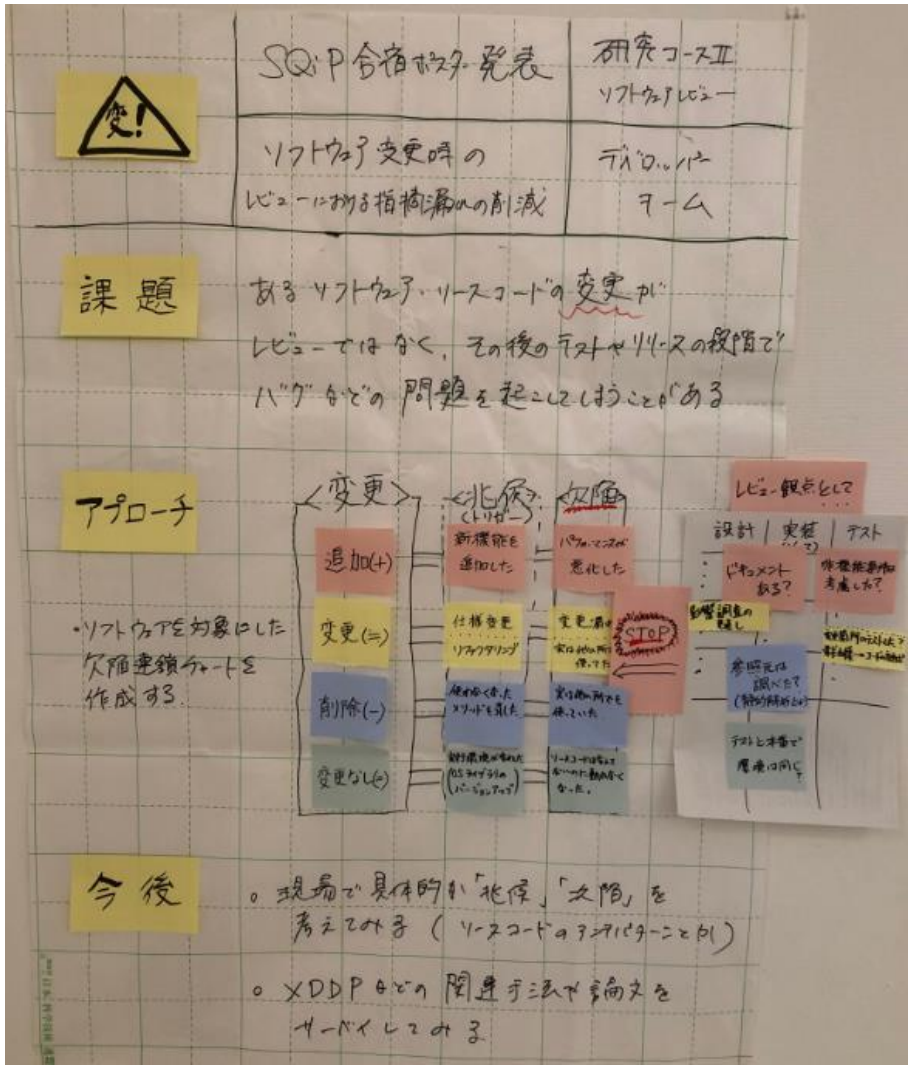
フォローアップ

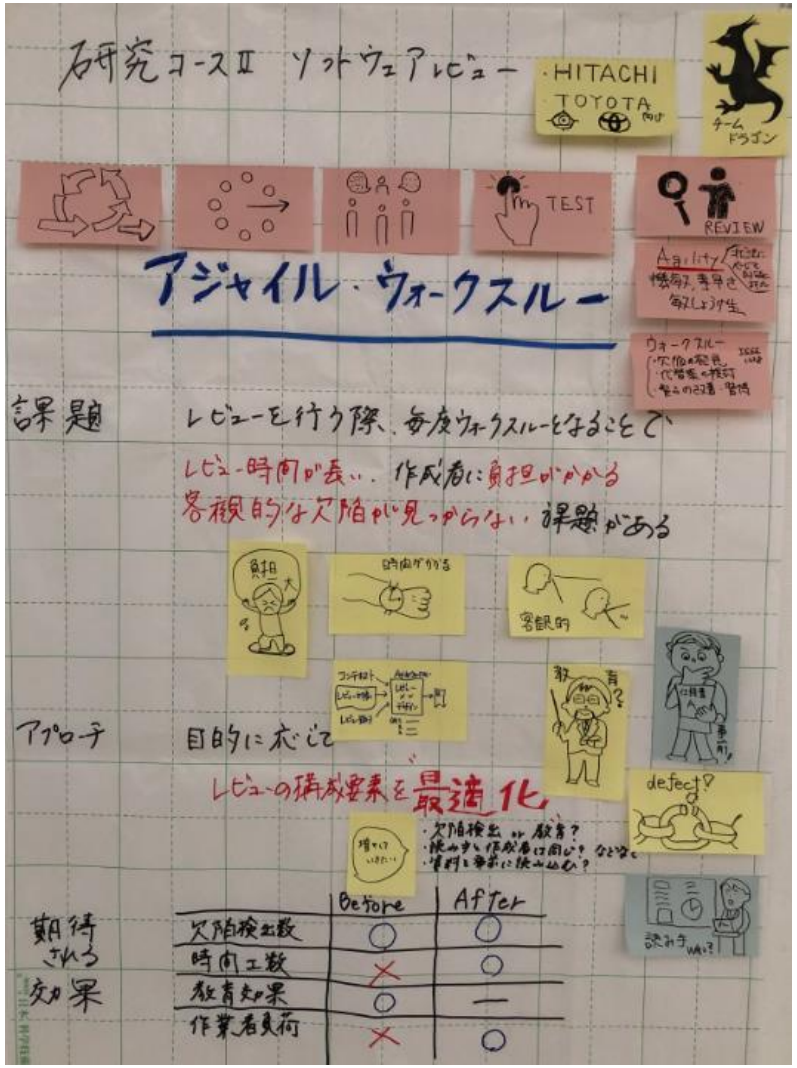
12

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

2019年度7月(合宿)

合宿成果物⇒20200221最終成果発表題名





その最終形は・・

レトロチーム

レビュー記録票を活用したレビュー振り返り手法

その最終形は・・

デベロッパーチーム

浅い知識や経験でも有効な指摘を行えるレビュー手法

その最終形は・・

オプティマイズチーム

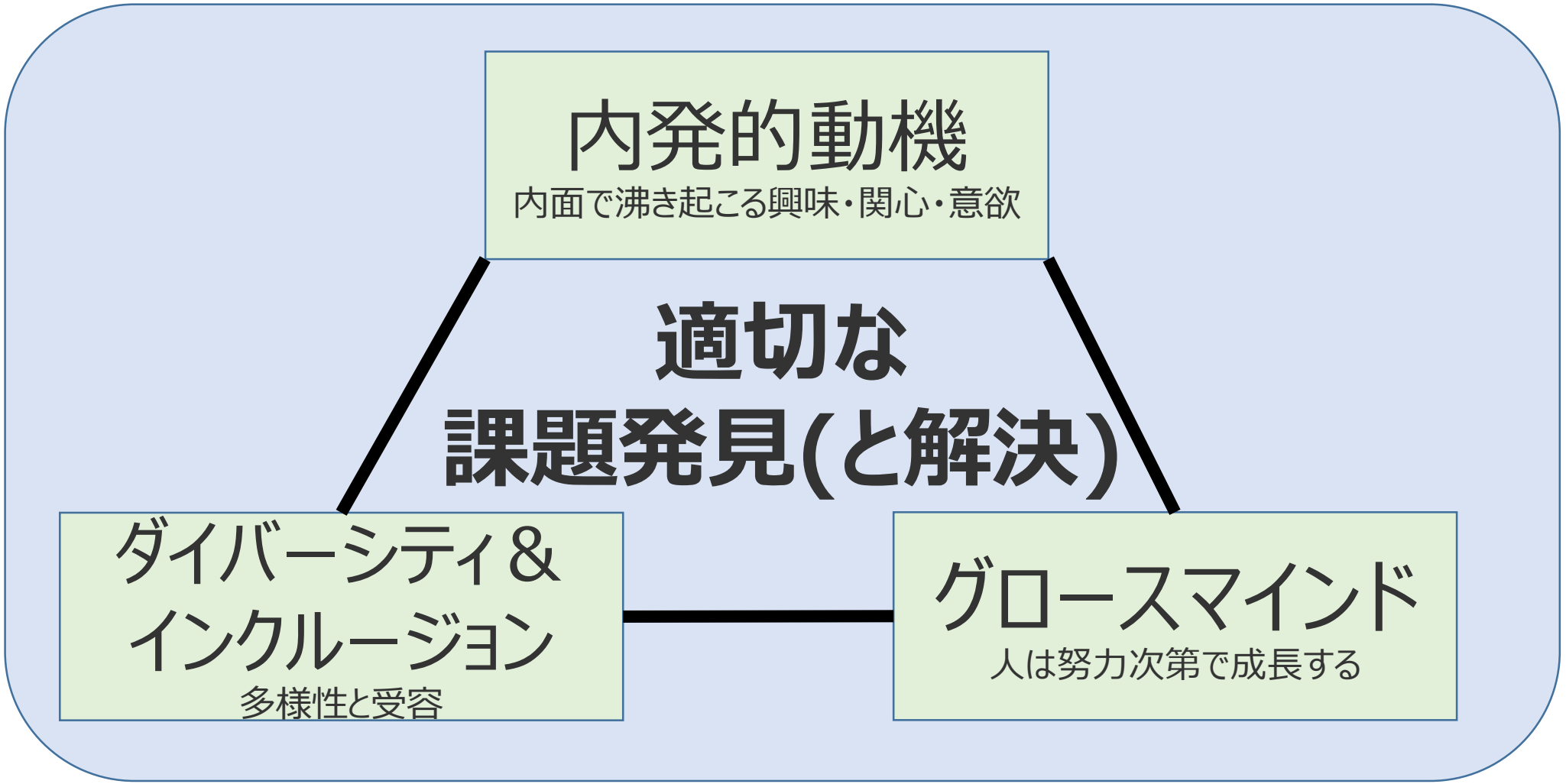
効果に応じたレビューの活動要素の選定

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPID実践～パフォーマンス向上のポイント

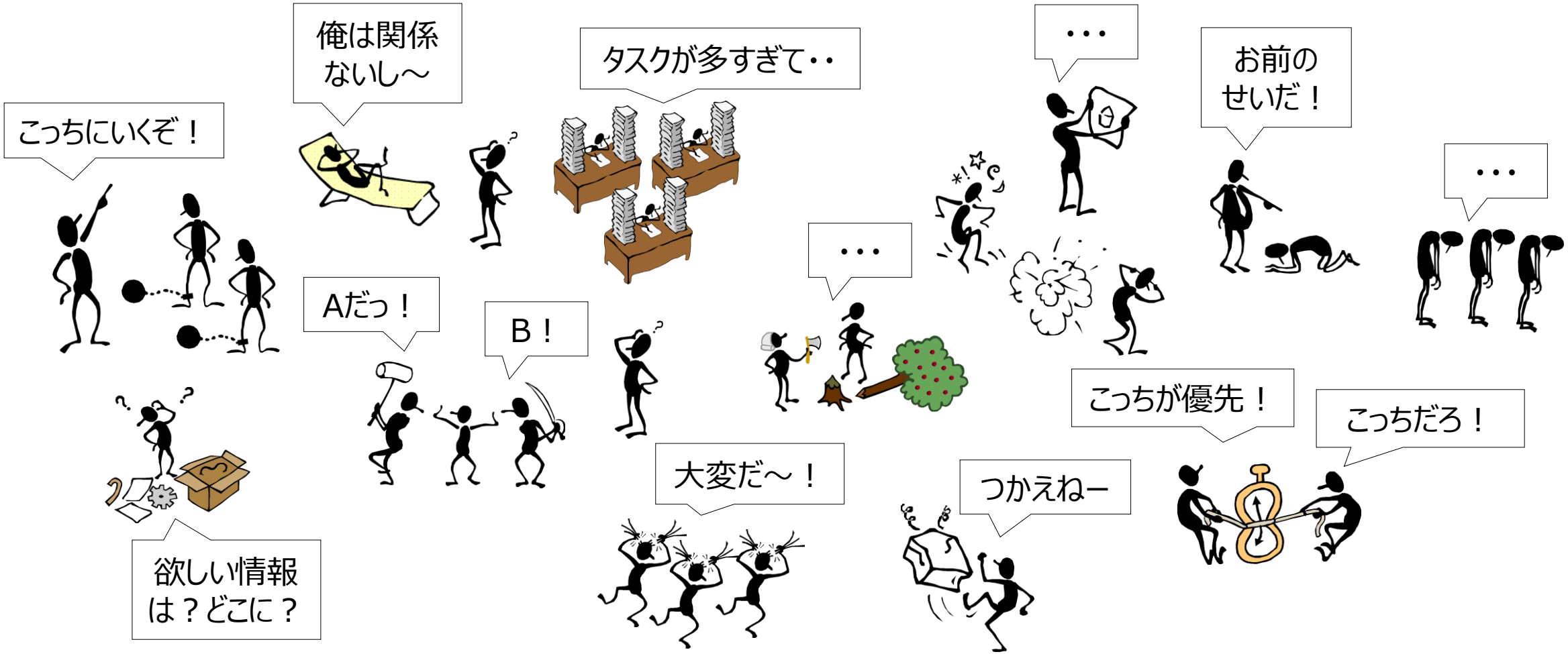
Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

パフォーマンス向上のためのポイント



Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

組織・チームには多くの問題が・・・・・・
しかし、その解決に使えるリソースには限りがある



Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

適切な課題発見

現状を構造的に理解して
っとも適切な課題を特定する

作り込み要因

流出要因

要求定義

現状分析結果がどこにも存在しない

要求なのか仕様なのかが不確かな記述形態

変更要求を処理する際に仕様への展開に一部失敗していた

レビュー対象の記述内容に反応するだけのアドホックレビューで見逃されていた

基本設計

複数の要求をそのまま設計仕様化し、詳細化不足になっている

設計仕様のみ記述する形式となっている
(どの要求に対する設計仕様なのかは要求仕様書と突合しないとわからない)

レビュー対象の記述内容に反応するだけのアドホックレビューで見逃されていた

機能テスト

テスト設計後の仕様変更への対応がコード修正のみとなっていた

基本的に*CPM法*でテスト設計を行っていたが、テスト設計後の仕様変更発生箇所については未対応であった

FT検証観点が設計仕様を網羅できていない事項あり

FT設計結果レビューが未実施であった

STでFunction不具合が想定以上に残存

Function不具合が出続ける

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

適切な“課題”の発見

“課題”＝多くの問題の中に存在する“解決すべき問題”

- The right answer to the wrong problem is very difficult to fix.

間違った問題への正しい答えほど、始末に負えないものはない

Peter.F. Drucker

- 問題解決力とは、優れた解決策を出す能力ではない。

本当の問題に気づく能力のことである。

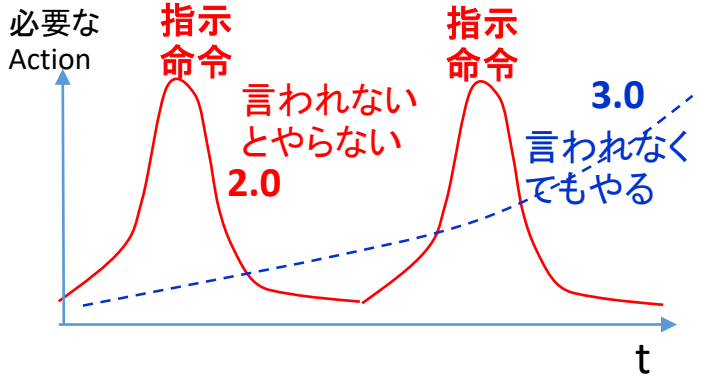
Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

「モチベーション3.0」ダニエル・ピンク

- **モチベーション2.0**(アメとムチなど外発的動機付け)

から**モチベーション3.0(内発的動機付け)**という2つの動機付け）へ

- モチベーション2.0の管理で報酬を用意すると、管理される側はその報酬のために短期的にやるだけになり、「自律性(オートノミー)」を失う。
- アメとムチの致命的な７つの欠陥
  1. 内発的動機づけを失わせる。
  2. かえって成果が上がらなくなる。
  3. 創造性をむしばむ。
  4. 好ましい言動への意欲を失わせる。
  5. ごまかしや近道、倫理に反する行為を助長する。
  6. 依存性がある。
  7. 短絡的思考を助長する。



「ODC分析を導入しろ」「ODC分析を導入することになりました」は
モチベーション2.0？3.0？

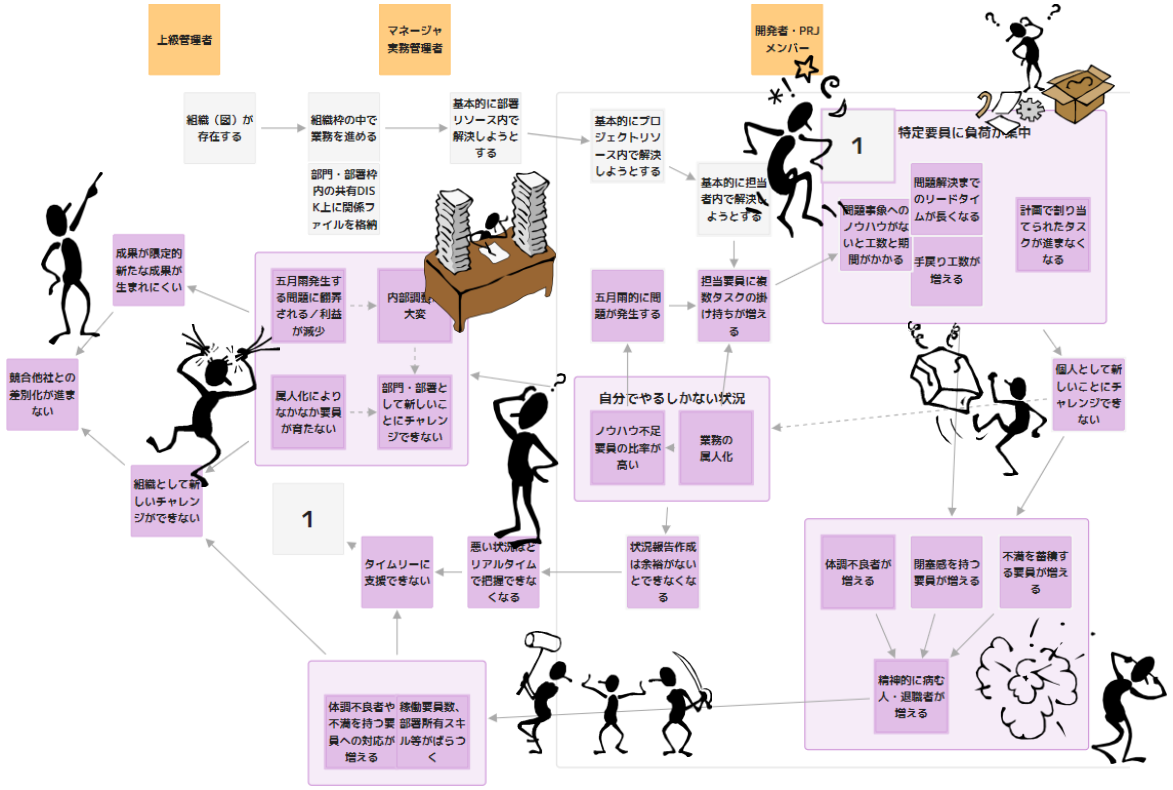
Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

内発的動機

SaPID～身近な困り事や問題を自ら見える化しつつ目指すゴールと連携させて解決していく

- モチベーションの源泉
  自分の身近な困り事や問題が解決するのはうれしい。

チームワークを高めながら自分事の範囲を徐々に広げていく



取り組み過程で発見する／あらためて認識すること

□普段の仕事で目指していることは何かそれはどのような価値があるのか
□自らの仕事の過程や結果が周囲に思わぬ影響を与えている
□他のメンバーが何の仕事をどのように実践しているか／どんな状態か
□自分の知らないところでいろいろなことが発生している／それに自分も加担している場合がある

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

Microsoft社の経営方針
ダイバーシティ＆インクルージョンとグロースマインド

ダイバーシティ＆インクルージョン
多様性と受容

人がみな「違う」ことを大事に

□「自分だけよければいい」ではなく、その人がその人らしい能力を存分に発揮できる場を、みんなで助け合ってつくっていこうというマインド

□「相手のことを理解して認める」エンパシー力（自分とは異なる相手の意見を知的に理解する力）で物事を進める

グロースマインド
人は努力次第で成長する

人は変わり、成長するものだ

□いまはできなくても、勉強して変わろうという人を応援するとともに、他人を応援しサポートする人を評価する

□みんなで助け合いながらお互いに成長できる働きやすい場に

「他人を応援する人が評価された結果…」44歳で転職したエンジニアが実感したマイクロソフトが“女性に働きやすい”理由　より

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

チームを成功へと導く5つの鍵

Google re:Work掲載 GoogleとAP通信社との共同研究成果より

信頼性 (Dependability)	構造と明瞭さ (Structure & clarity)	仕事の意味 (Meaning of work)	インパクト (Impact of work)
任せると着実にプロセスを進めるメンバーがいる	仕事のゴールと役割、プランを明らかにする	仕事がどのような個人的な意味を持つのかを知る	仕事がどのような影響を社会に与えるのかを知る

心理的安全性(Psychological safety)

4つの不安＝無知だ・無能だ・邪魔してる・ネガティブだ

【個人ができる取り組み】

(1)仕事を実行の機会ではなく学習の機会と捉える。
(2)自分が間違うということを認める。
(3)好奇心を形にし、積極的に質問する。
(以上はエドモント氏の提言)

【管理者ができるアプローチ】

1.発言機会を平等に与える
2.競争よりも協力を
3.ポジティブ思考を意識する
4.上司が部下を尊重する
5.付加価値の高い1on1を実施する
6.チームで新入社員を支援する
7.評価方法を見直す
8.風通しの良い組織を作る

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

政治でもスポーツでも『熱狂』は危険なものだ。**その人が信じることだけを正しいとする盲目**につながり、あらゆる疑問を覆い隠す。

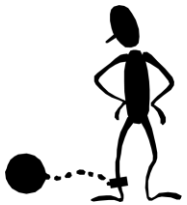
それは賢さとは対極にあるものだ。

ウルグアイ　ホセ・ムヒカ元大統領

専門家バイアス

出典：https://this.kiji.is/916396302473134 11　琉球新報

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved



専門家バイアスから脱却するには？
＝自分を成長させるには？

☑まずは自分は専門家バイアスにかかっていると自覚する。

☑異なる見方、意見から学び、自らの既存知識と併せて新しいモノゴトの捉え方、考え方に更新し、より適切な思考・行動へと統合する。

↓

ふりかえりが有効

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

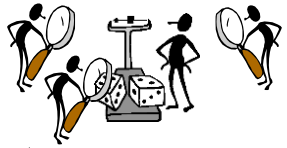
私が感じている“ふりかえり”のよい効能

ふりかえりカンファレンス2021～ふりかえりの傾向と対策「ふりかえりのふりかえり」から作法を学ぶ　より

☑相互理解

自分が、それぞれのメンバーがどのような人なのか、どんな状況なのかを共有できる。

☑チーム学習



他者の気づきを自分の気づきにできる。自分の実務(経験)だけではわからないことを他者から学べる。チームとして学ぶことができる。また、対話やファシリテーションを通じてモノゴトの適切な捉え方や伝達方法を浸透させることができる。

☑合意形成と実践促進

チームとしての合意形成が容易になる。チームで決めたことの実践が促進される。

☑価値観共有・共創～そして浸透

チームの価値観を共有したり、新しい価値観やより適切な価値観を一緒に作り上げる、それを浸透させることができる。

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

自分自身への気づきを高める法

「スーパーエンジニアへの道　技術リーダシップの人間学」第七章

- 自己変革に成功するのに十分なだけの動機づけを持っているかを調べるテスト～たったいまから三箇月間、個人的な日誌をつけるために毎日五分使うこと
- 日誌には何を書くべきか？

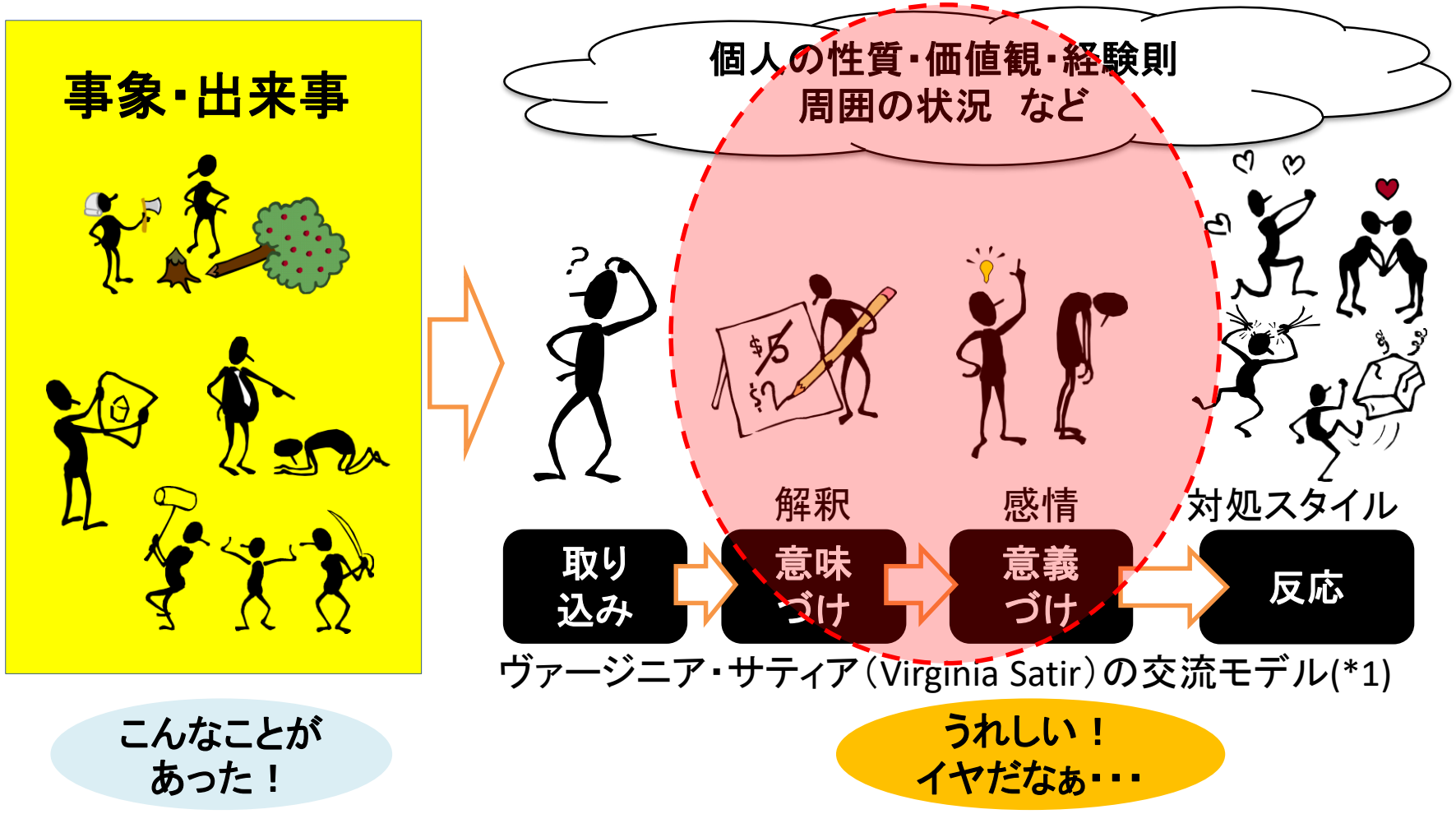
「自分について書く」

- **事実**：自分に何が起こったか？（客観的に）
- **感情**：自分はそれにどう**反応**したか？
- **発見**：それにより何を学んだか？

- 日誌は**自分自身について学ぶため**につける。たいていの場合、**学ぶのはずっとあとになってその項目を読み直したとき**です。

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

人間のモノゴトの取り込み～反応まで



*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

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

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。

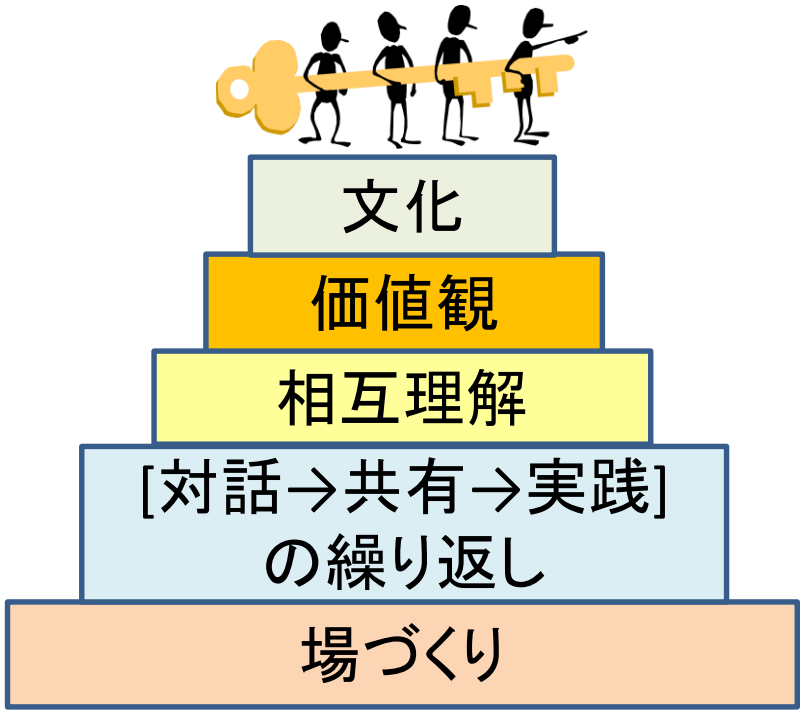
Gerald.M.Weinberg

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。

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

相応しい価値観→文化は大事なチームの礎

- ふりかえりを**適切に、繰り返し実践**することで、相応しい価値観に基づく運営が浸透すればそれが“**チームの文化**”になる。
- **チームの文化がチームのパフォーマンスを左右する。**



ふりかえり
による基盤づくり

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

カーリングチーム：LocoSolare（ロコソラーレ）

- 本当に強いチームとは？

https://locosolare.jp/

「ピンチが来ても復活できるチーム」

ロコソラーレ代表理事　本橋 麻里さん

https://www3.nhk.or.jp/news/html/20220108/k10013416951000.html

「勝ち続けるチームではなく、負けてから這い上がれるチーム」

ロコソラーレ　サード　吉田 知那美さん

https://real-sports.jp/page/articles/6296502705397440 59

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

本音のコミュニケーション

「カーリング 本橋麻里 追い求めた本音のコミュニケーション」他　をベースに編集

- カッコつけず素直に言葉にする＝そのままの自分で
  - メンバーの素養を**そのまま受けとめられるメンバー、チームは強い**
  - 腹を割って話せるチームメンバーがいる（信頼感／お互いを尊重する）
  - 意見が言える・本音でぶつかり合える～エネルギーをもらえる、人生が豊かになる
- メンバーに興味を持つ＝**仲間を知る**～そのための“**傾聴**”と“**対話**”
  - どのようなコトやモノが好きか？等、メンバーの心が動く瞬間を知る
  - 相手の気持ちを引き出す質問等会話のキャッチボール（対話）で掘り下げる
  - 一見くだらない、他愛のない会話に大事なヒントがある
  - **互いに弱さを見せあえる→ピンチの時こそ支え合う**

心理的安全性の前提”受容”

本当の自律＝相互依存

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

カーリングの理念・文化

- 相手を見下さない。
- プレーを妨害しない。相手の負けを喜ばない。
- 審判はいない／卑怯なことはしない。
- 不当に勝つなら負けを認める。
- ギブアップではなくコンシード（相手に“譲る”）。
- 勝ったチームがシートの掃除を行う。
- 勝負が終わったら対戦相手と一緒にお互いの戦いをふりかえる。

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved

まとめ

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

いかがでしたでしょうか？

今回は

- SaPIDとは何か？
- SaPIDを活用した研究の進め方
- 　Case：ソフトウェアレビュー研究
- SaPID適用事例
- SaPIDを活用したSQiP研究会マネジメント
- SaPID実践～パフォーマンス向上のポイント
- まとめ

についてお伝えしました。

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

【参考】SaPID実践道場

https://www.softwarequasol.com/training/sapid-dojo/

- SaPIDは**「聞けばわかる、でもやってみると難しい」**ノウハウです。（とよく言われます）
- さまざまな状況で日々実践している人に直接教えてもらうのが能力を高めるために最も効果的。ということで2024年度から「SaPID実践道場」を開くことにしました。
- ひとり一人の状況に合う実践を指向するため**「マンツーマン形式」**で運営します。（受講者数制限あり）
- 受講条件などを含めて直接確認したい方はご連絡ください。

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

SaPIDで
パフォーマンス改善沼に飛び込もう！

今後研究を進める際の
参考になれば幸いです

お疲れさまでした！

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

参考文献

- 「ソフトウェアプロセス改善手法SaPID入門」 安達賢二
- Software Quasol SaPID® Ver4.0 (SaPID4.0) https://www.softwarequasol.com/sapid/
- 「ワインバーグのシステム洞察法」 G.M.ワインバーグ
- 「学習する組織――システム思考で未来を創造する」 ピーターMセンゲ
- 「プロジェクトマネジメント知識体系ガイド（PMBOKガイド）第7版＋プロジェクトマネジメント標準」 PMI日本支部
- -the d.school bootcamp bootleg
  https://dschool.stanford.edu/resources
- Design Thinking for Educators Toolkit,IDEO,2011
  https://page.ideo.com/design-thinking-edu-toolkit
- ソフトウェアテストシンポジウム2019新潟（JaSST2019新潟）「レビューのキキメ　～あなたのレビューは何のため？」
  http://www.jasst.jp/symposium/jasst19niigata/pdf/S1.pdf
- ソフトウェアシンポジウム2023仙台
  「ソフトウェアレビュー研究結果の認知拡大と適用促進」活動状況報告（事例報告）
  https://www.sea.jp/ss2023/download/16-ss2023.pdf

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

参考文献

- SPI Japan2015「自律型プロジェクトチームへの変革アプローチ事例」　安達賢二
  http://www.jaspic.org/event/2015/SPIJapan/session3C/3C-3_ID012.pdf
- SPI Japan2020「FRI(Factor-Risk-Influence)モデルによるリスク構造の見える化」　安達賢二
  http://www.jaspic.org/event/2020/SPIJapan/session6/6-3_ID008.pdf
- ODC分析研究会オープンセミナー「ODC分析の概説」
- 「[入門+実践]要求を仕様化する技術・表現する技術 -仕様が書けていますか?」 清水吉男
- 成功循環モデル　マサチューセッツ工科大学ダニエル・キム
- ふりかえりカンファレンス2021発表　ふりかえりの傾向と対策 「ふりかえりのふりかえり」から作法を学ぶ」 安達賢二
  https://speakerdeck.com/kitanosirokuma/hurikaerifalseqing-xiang-todui-ce-hurikaerifalsehurikaeri-karazuo-fa-woxue-bu
- 「モチベーション3.0」 ダニエル・ピンク
- 「他人を応援する人が評価された結果…」44歳で転職したエンジニアが実感したマイクロソフトが“女性に働きやすい”理由 https://crea.bunshun.jp/articles/-/46227?page=2&utm_source=crea&utm_medium=direct&utm_campaign=nextpage_link

Copyright © Kenji Adachi@Software Quasol, All Rights Reserved

参考文献

- グーグルが発見！成功するチームに共通する「心理的安全性」のつくり方
  https://gendai.media/articles/-/66539?page=1&imp=0
- ＜社説＞ムヒカ氏初来日　戦争なき世界実現考えたい　琉球日報
  https://this.kiji.is/916396302473134 11
- 「スーパーエンジニアへの道　技術リーダシップの人間学」 Gerald.M.Weinberg
- カーリング 本橋麻里 追い求めた本音のコミュニケーション
- ロコ・ソラーレ、銀メダル最大の要因は「勝つか学ぶか」。負けるたび強くなる“成長の本質”　［カーリング］
  https://real-sports.jp/page/articles/62965027053974 4059/

Copyright © Kenji Adachi@Software Quasol , All Rights Reserved