

レビュー指摘のカラクリを解き明かして
狙った指摘をするノウハウ
を持ち帰ろう
【上流フェーズ編】

Software Quasol
安達 賢二

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/>

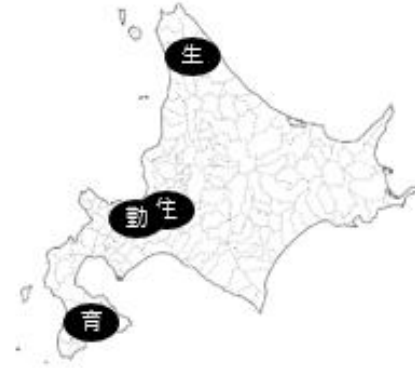
【経歴】

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】

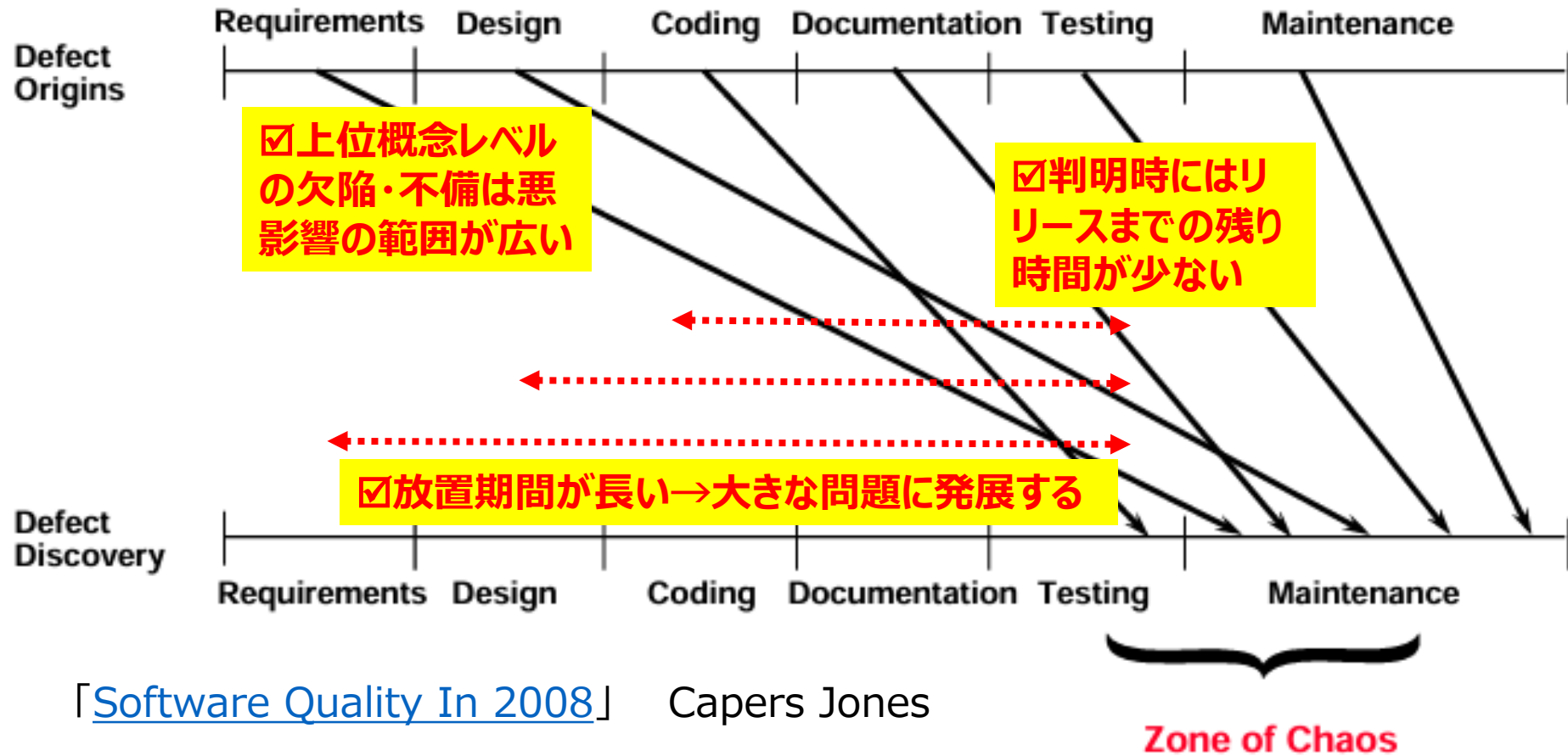
NPO法人 ソフトウェアテスト技術振興協会（ASTER）理事
JSTQB（テスト技術者資格認定）技術委員
JaSST（ソフトウェアテストシンポジウム）北海道
2006-2009実行委員長 2010-2018実行委員 2019～2022サポーター
JaSST-Review（ソフトウェアレビューシンポジウム）実行委員
JaSST-nanoお世話係
ASTER研究事業「レビュー体系化」メンバー
テスト設計コンテスト本部審査委員(2015-2017)
SEA（ソフトウェア技術者協会）幹事・北海道支部メンバー
SS（ソフトウェア・シンポジウム）プログラム委員
第33-39期SQiP研究会レビュー分科会ア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



チュートリアルのねらい

- レビュー観点の重要性を理解し、レビュー目的とその実現に必要な観点の設計方法（の一つ）とその効果を把握する。
- このアプローチを現状のレビューにどのように適用するのがよいかを明確にする。

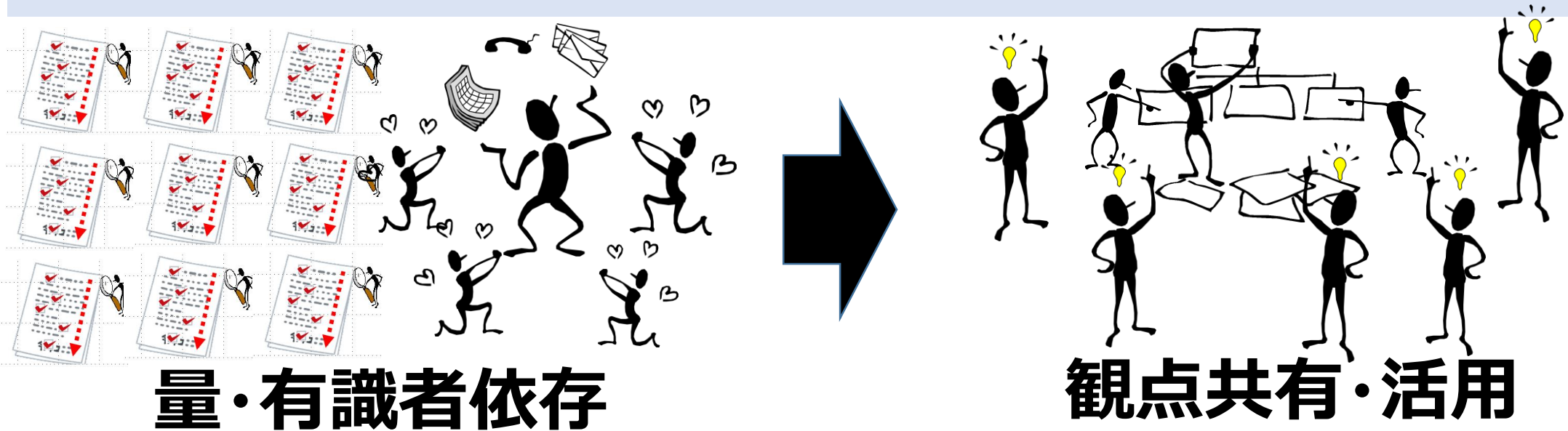
上流フェーズ成果物を対象としたレビューでの欠陥・不備の見逃しがソフトウェアプロジェクトに与える悪影響は大きい



上流フェーズレビューの欠陥・不備の見逃し防止・緩和はプロジェクトリスクを低減する

観点活用レビューが目指していること = **量**から**質**への転換

「**誰が、どのくらい確認したか？**」から
「**何を、どのように確認したか？**」
へのシフト！



想定する受講者と期待効果

- 想定する受講者
 - いつもレビューで見逃しが多いと実感している方
 - レビューを実施しているが、効果が実感できない方
 - レビューの結果が有識者の出席有無に大きく依存するチームのメンバー
- 期待効果
 - より有効な欠陥を検出する能力の向上
 - ぼんやり、なんとなく実施するレビューから狙い撃つレビューへの第一歩
 - 業務知識に頼らず欠陥や不備が指摘できる領域の拡大
 - レビューの成果を有識者に頼りすぎる状態からの脱却
 - 要求仕様、設計書作成時の考慮事項や注意事項の導出力向上
 - テスト設計との連携によるQAアプローチの足掛かりに



実施概要

以下の内容を自ら実践することで、レビュー観点設計方法（の一つ）を学び、その効果を実感する。

- レビュー対象成果物をレビューして指摘事項から観点を逆算する。
- 設計したレビュー観点に基づきレビューを実施する。
 - レビュー対象成果物に必要なレビュー観点を導出・体系化する。
 - レビュー観点に基づきレビューを実施する。
- 観点ベースレビューの効果を(いつも通りのレビュー結果と比較し)評価する。
- ふりかえりにより実務への適用方法を検討する。

チュートリアルで行うこと

レビュー観点に基づく実践を
制限時間の中で部分体験する



レビュー観点に基づく実践を
完遂するまで体験する





時間が来るまで実践／時間が来たら終了
その中でエッセンスを獲得するためのワークです。
あらかじめご了承ください

- 普段のレビューでは、レビュー対象すべてをレビューするのが当たり前ですが、本日は時間が限られているため、

時間枠内で実践する

→ **時間が来たら終了**

→ **試行体験から得られたことをふりかえりに出力**

→ **ふりかえり結果共有**

という流れで進めます。

チームビルディング：10分間

- ・ブレイクアウトルームにて以下の対応を行ってください。

①メンバーの自己紹介[1人1.5分×人数]

[氏名・所属][最近のハマり事]

[普段実施しているレビューの目的]

②最初のワーク時の役割を決める

[リーダー][タイムキーパー][記録係][盛り上げ役]

リーダー

タイム
キーパー

記録係

盛り上げ役

突然ですがレビューを実施してください

- 指定するレビュー対象を15分間各自でレビューしてください。
- 欠陥や不備、確認必要事項があれば行番号とともに記録してください。

レビュー結果記録様式

観点導出アプローチ番号： A			担当者：	
N o.	指摘箇所	指摘グレード 重・中・軽	指摘内容	
1				
2				
3				
4				
5				

レビュー目的・観点分析

- 最初のレビュー結果のうち任意の1件を選択し、「**レビュー目的・観点分析表**」を活用して分析してみましょう。(すぐ終わった方は2件目、3件目と実施)

【時間：15分間】

分析が早く終わった方は、グループ内の方達と分析結果を比較してみてください。

レビュー目的・観点分析表

(4)観点はどのような目的を達成するためのもの？	(3)どのような観点でレビューした結果？	(2)どのように調べた結果？	(1)レビューで検出した欠陥・不備の内容

レビュー観点分析例

(4)観点はどのような目的を達成するためのもの？	(3)どのような観点でレビューした結果？	(2)どのように調べた結果？	(1)レビューで検出した欠陥・不備の内容
使用性・保守性が確保されていることを確実にする	ドキュメント内、および画面・機能間整合確認（一貫性・整合性） システムを一貫した構造や用語使用で構築するため	類似画面を洗い出し、同じ意味のオブジェクトや説明を目視で比較確認→表現や形状が不一致の場合は指摘する	P1では「登録」ボタンなのに、P3では「書き込み」ボタンになっている



(2)～(4)

左に一つずつシフトしながら回答する



(1)

検出した欠陥・不備を転記

レビュー結果記録様式

観点導出アプローチ番号： A			担当者：	
N o.	指摘箇所	指摘グレード 重・中・軽	指摘内容	観点
1				
2				
3				
4				
5				

ふりかえり

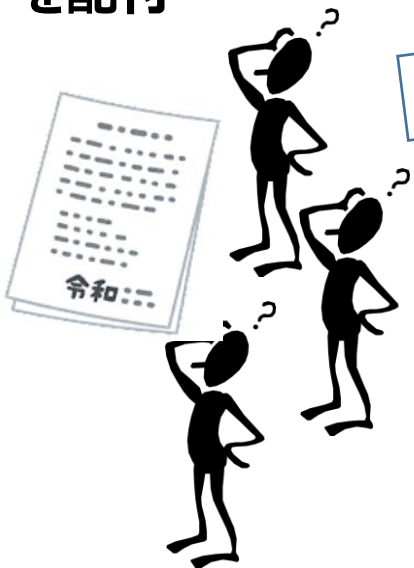
実施内容に対してふりかえり、結果を残しましょう

Keep:わかったこと・気づいたこと・感じたこと など	Try:今後の対応や方向性
Problem:現状の問題点や課題	

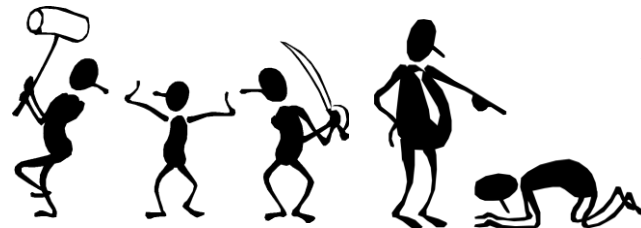
事前解説～観点に基づくレビュー

典型的なレビューの状態

レビュー会議で
初めてレビュー対象
を配付



ジャイアン独演会・あら捜し・
横道逸脱・人格否定・いざこざ

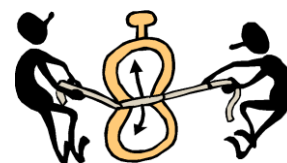


書かれていることに反応し、
気が付いたことを指摘

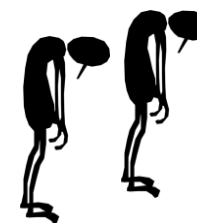
有識者がいないと軽微
で、表面的な指摘ばかり



多忙につき
レビューを省略



モチベダウン



レビューの効果が
実感できない



見逃した欠陥が
あとになって爆発



こんなことになっていませんか？

先日障害発生したからレビューは全員でやれ！

多重度を上げると漏れがなくなるんじゃないかレビュー

大勢でレビューしたからもうバグはないんだよね？

すべてレビューしたのになんでこんな問題が発生するんだ！

やっとできあがったのでレビューをお願いします

忙しいのに・・・大量の赤入れが必要な成果物って・・・

対象がすべてできあがらないと実施しないレビュー

こっ、これ全部直す？

レビューで欠陥がないことを証明できると勘違い

作成者

レビュー担当

作成者

レビューへの誤認識や不適切な行動と結果

- レビューとは、レビュー会議でレビュー対象をチェックして指摘すること。
- レビューでは、対象を知るための情報源（例：仕様書やコード）に書かれていることを確認すればよい。
- レビューで十分な確認を行うには、レビューアの人数を多くすればよい。
- レビューチェックリストに記載されている項目を確認すれば終わりだ。
- レビューの成果は業務知識豊富な有識者が出席出来るかどうかで決まる。



その結果・・・

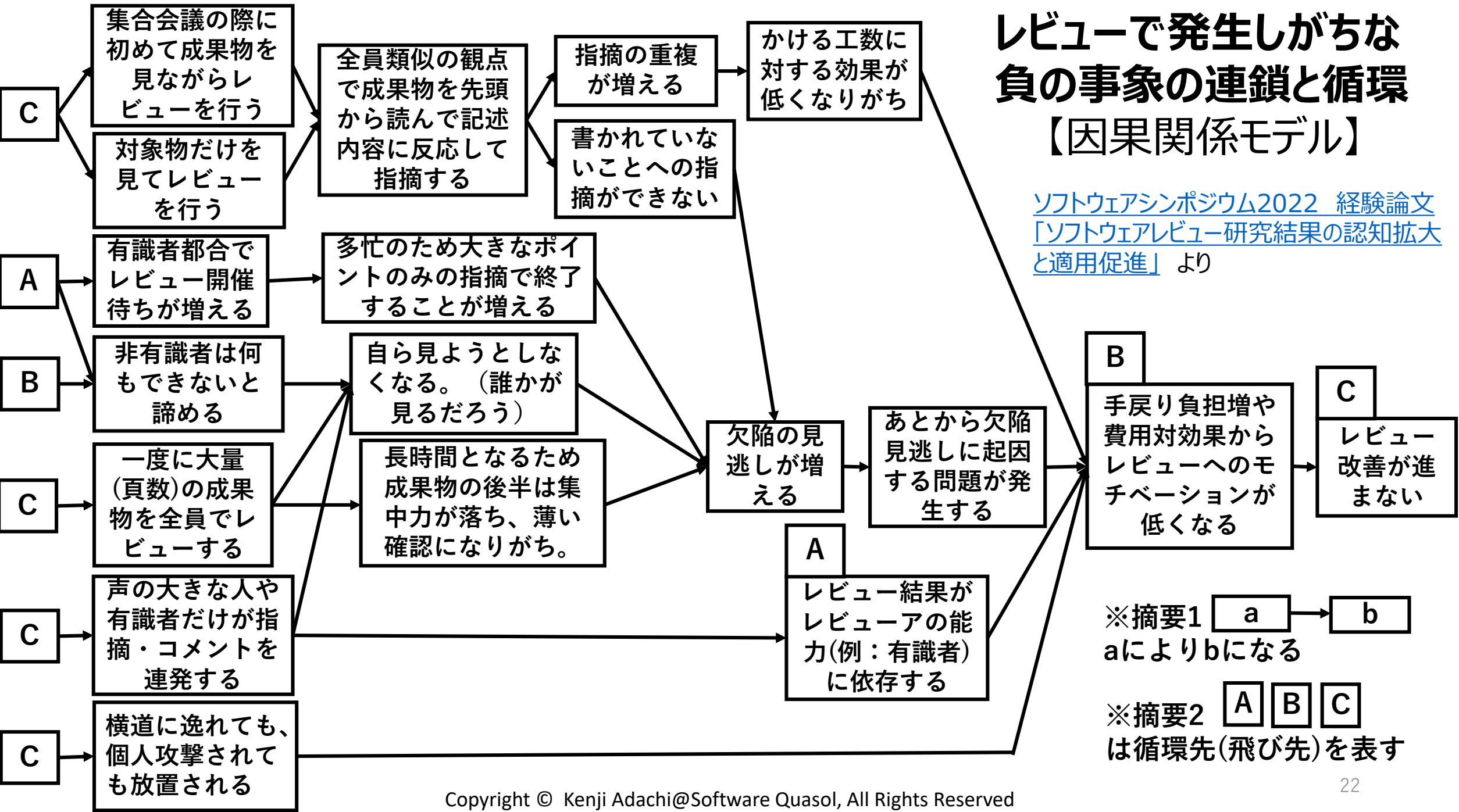
- レビューでの見逃しが多く、後工程で問題が発生する。
- もうあの人の独演会レビューには参加したくない。
- レビュー時間がムダに長く、ダメ出しが多くて気が滅入る。
- 業務知識が重要→レビュートレーニングなんて必要ない。
- レビューの効果が実感できない。

**理解不足や誤認識は
不適切な行動を生み
イヤな結果を引き寄せる**

レビューで発生しがちな負の事象の連鎖と循環

【因果関係モデル】

[ソフトウェアシンポジウム2022 経験論文「ソフトウェアレビュー研究結果の認知拡大と適用促進」](#) より



JaSST2022東京ワークショップ参加者が 感じていたレビューの問題点

レビューでの**見逃しが
多い**(時支出来ていない
ことも?)

レビュー**指摘の出来が
不安定**

**工数がかかる
レビューイ理解度(レ
ビューアの伝え方)**

**人によってレビュー
の観点と粒度がバラ
バラ** / ほぼ機能してい
ないレビューもある

レビュー**成果物の質**
発言・指摘が毎回同じ人
**レビューMTGの進
め方**
レビューの効果が可視化で
きてない

**レビューアの勘と経
験に頼ったレビュー
になってしまっている。**

途中から参加したプロダクト
について**突っ込んでい
いのか悪いのかわか
らない**

**必要以上に時間が
かかっている
レビューイの心構え
(受け身)**

**経験則に頼りすぎて
いる**

レビューイとレビューア間の関係
性が薄くて**協力関係にな
れなさそう**なとき困る

レビュー以前に、何がわから
ないのか探す習慣がない
非形式な**レビューをカジュアル
に**やってみるなど、取り組みが
できていない
うまくいってない感を解消したい
とおもえる、前向きな気持ちを
育てたい
レビューを身近なものにしたい

[ソフトウェアレビューシンポジウム2022東京 ワークショップ
「そゆことね！よくわかるレビューテクニック～明日から使え
る技術をSQiPレビュー研究会からあなたに～」](#) より 23

レビュー観点設定～狙い撃ちレビューのイメージ

あらかじめ対象
の内容や構造、
記述レベル等を
ざっと把握する



それぞれのレビューアが
観点に沿って確認

観点A

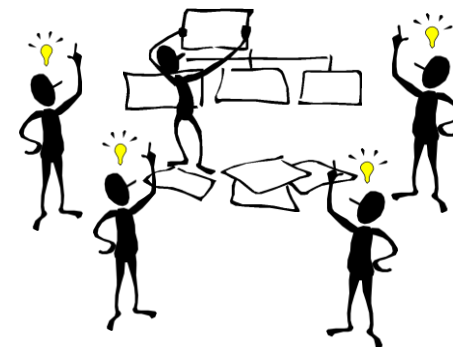
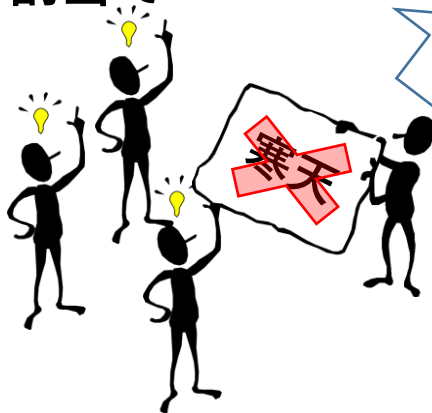
観点B

観点C

観点をそれぞれの
レビューアに
割当て

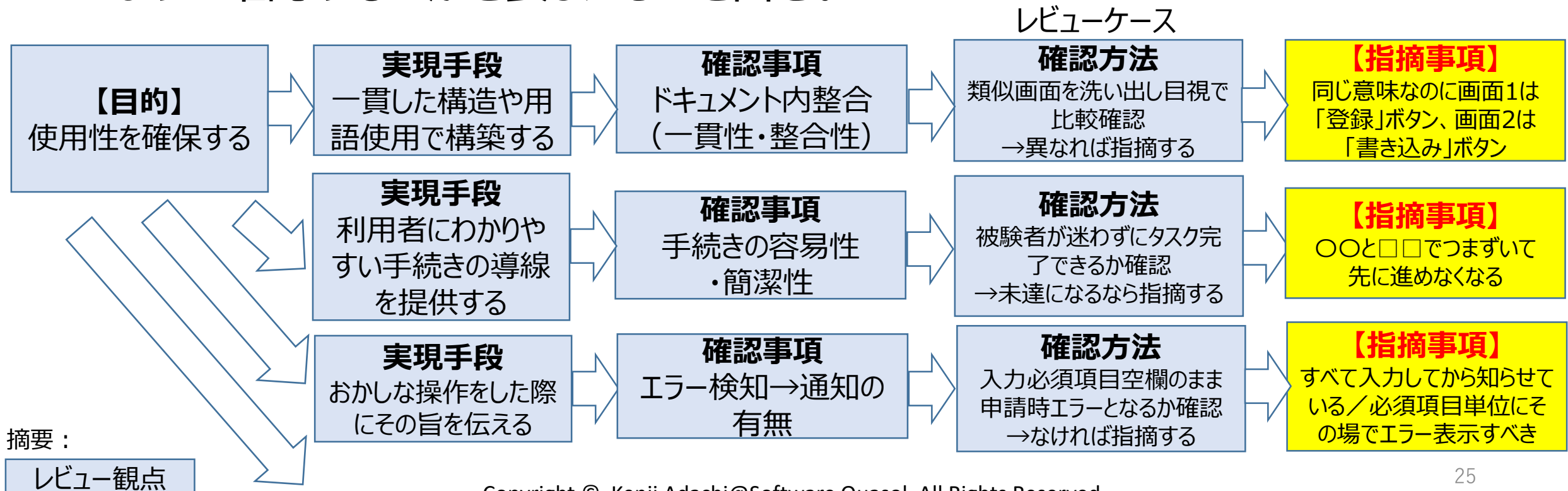
レビュー結果を評価し、
目的達成できているか
等をふりかえる（以降
のレビューに繋げる）

レビュー目的や制約条件
に応じて必要な観点を洗
い出し、どのように確認
するかを事前に整理する



レビュー観点とは？

レビューの意図や目的を段階的に詳細化したもの。
レビュー目的を達成するための、レビューアによるレビュー対象の見方、レビューで欠陥を見つけるために集中して着目する対象成果物の側面。さらに何を、どのように確認するのかを表したものを含む。



レビュー観点とは階層構造



抽象度の高いレビュー観点 (抽象的)

人によってはその意味や内訳が異なる可能性がある
【ハイレベルレビューケース：HRC】

粒度が大きい

【ミディアムレベルレビューケース：MRC】

抽象度が低いレビュー観点 (具体的)

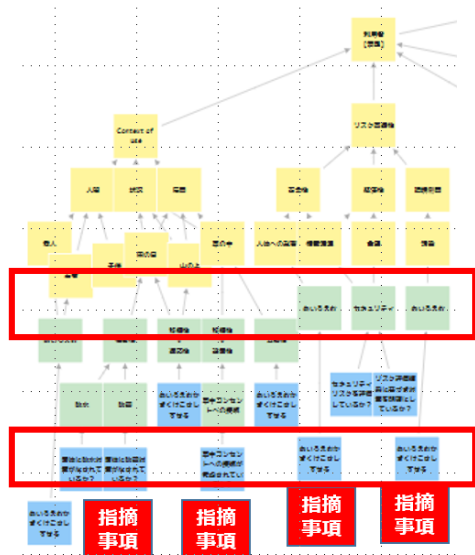
誰がチェックしても同じ結果が導ける可能性が高い
【ローレベルレビューケース：LRC】

粒度が小さい

人によって適切に実践できる観点粒度が異なる

レビューアの状態で観点粒度を調整する

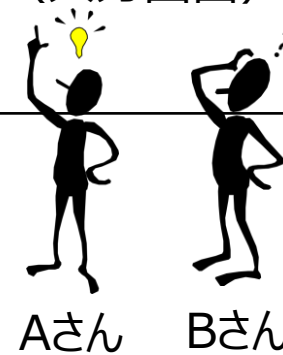
利害関係者	対象システムにおける主な活動	関心事	必要なレビュー観点・対象	レビュー観点の具体化
実利用者	システムに必要情報を入力して、ほしい結果を得る	誤入力情報がそのまま処理されないか？	入力ミス検知機能があるか？（入力画面）	☐ 必須項目記入漏れの検出 ☐ 入力不可文字の検出 ☐ 日付、時間範囲外の検出



Aさん

Bさん

OK! この場合だと
☐ 必須項目記入漏れの検出
☐ 入力不可文字の検出
☐ 日付、時間範囲外の検出
 ができればよいね!



Aさん

Bさん

ん～この場合だと
☐ 必須項目記入漏れの検出
 は思いつくけど他には何ができ
 ればよいのかな???

OK!

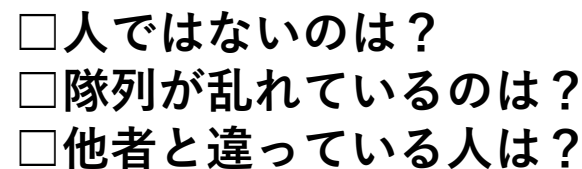


Bさん

なぜレビュー観点が必要なのか？

No.	内容
1	“観点”があれば雑多な情報群の中から <u>集中して該当を直接探し当てられやすい</u>
2	“観点”がない場合、 <u>書かれていることだけに反応した指摘に終始してしまう</u>
3	有効な指摘事項など <u>優先度の高い欠陥・不備を見つけやすい</u>
4	“観点”がない場合、メンバー間での <u>重複確認</u> や <u>誰も見ていない抜け・漏れ</u> が発生しやすい
5	“観点”があれば、 <u>メンバー間で確認する観点の分担が可能になる</u>
6	<u>有効指摘のノウハウを再利用しやすくなる</u> （ <u>有識者依存度を低減できる</u> ）

"観点"があれば雑多な情報群の中から**集中して該当を直接探し当てられやすい**

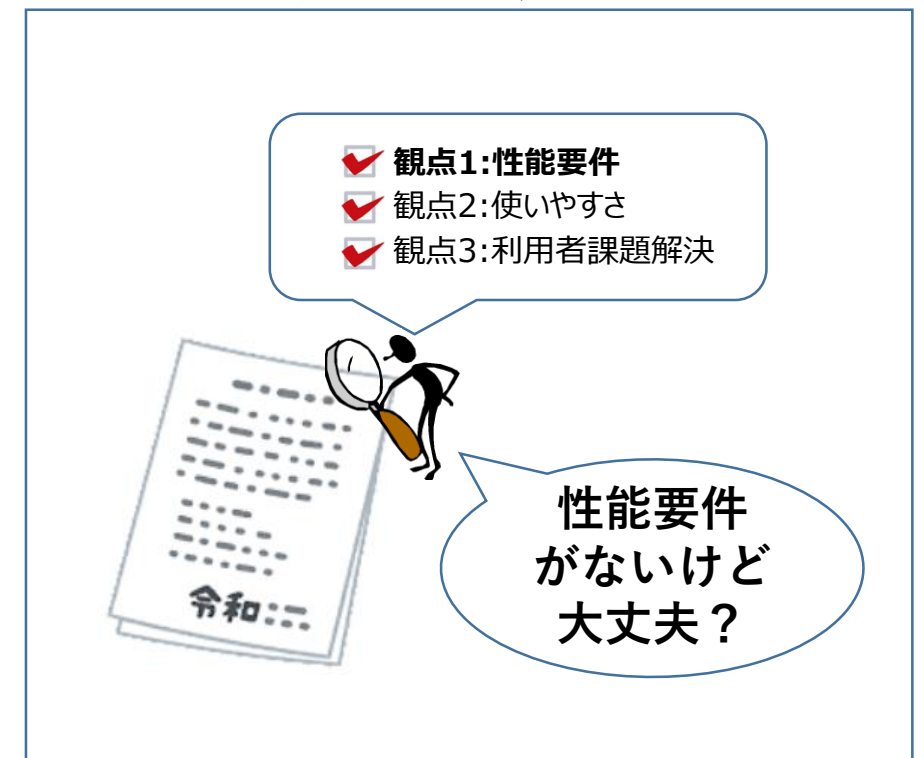
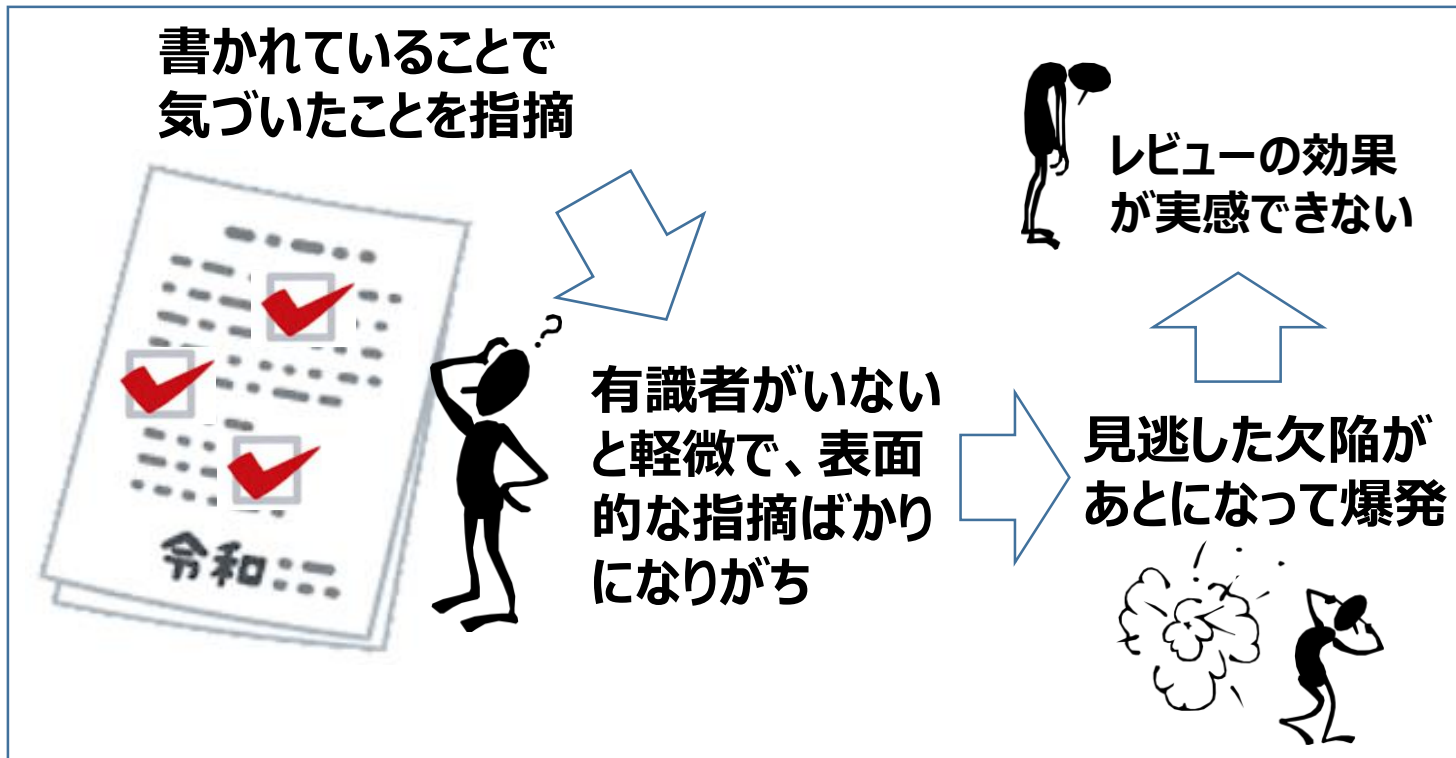


なぜレビュー観点が必要なのか？(2)

“観点”がない場合、書かれていることだけに反応した指摘に終始してしまう

対象を読みながらぼんやり行うレビュー

観点に基づくレビュー



レビューに期待されていること（テストでは実現が困難なこと）

書かれるべきなのに書かれていない／書いてあるけど必要ないことを見つける

なぜレビュー観点が必要なのか？(3)

有効な指摘事項など**優先度の高い欠陥・不備を見つけやすい**

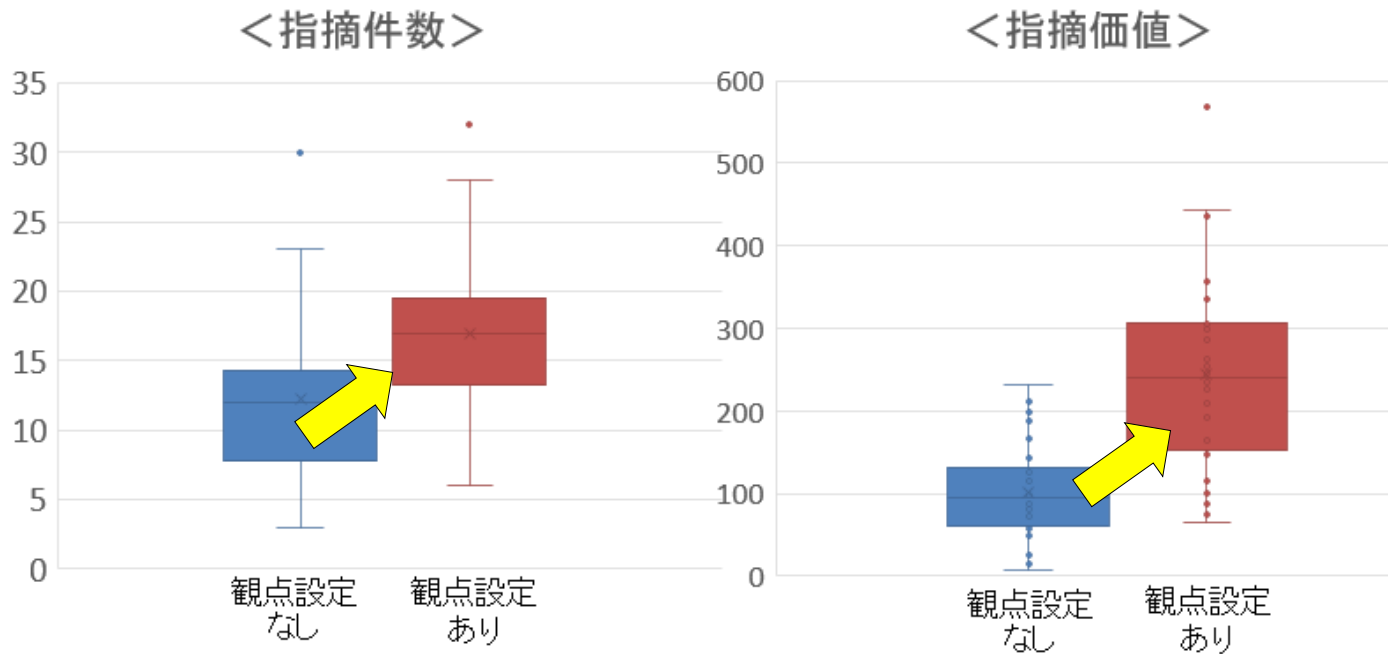
△改善前 (Before) ●改善後 (After)		発見可能Phase (想定)				計
		実装・UT	IT	ST・OT	C/O後	
		1	3	5	7	
検出効果	効果大 主対象: 要件抜け・誤り	5 				30 ↓ 335
	効果中 主対象: 機能上のバグ (誤植による)	3 			●が右上 (指摘価値が高い) に集中	75 ↓ 93
	効果小 主対象: 誤字・脱字・衍字 規約違反	1 				16 ↓ 8
	計	35→19	18→0	40→340	28→77	121→436

レビュー観点設計による典型的な指摘変化の例

△=アドホックレビュー指摘

●=観点設計レビュー指摘

2012～2020年に実施した全ワークショップ^o
(11社30チーム) 実績から算出

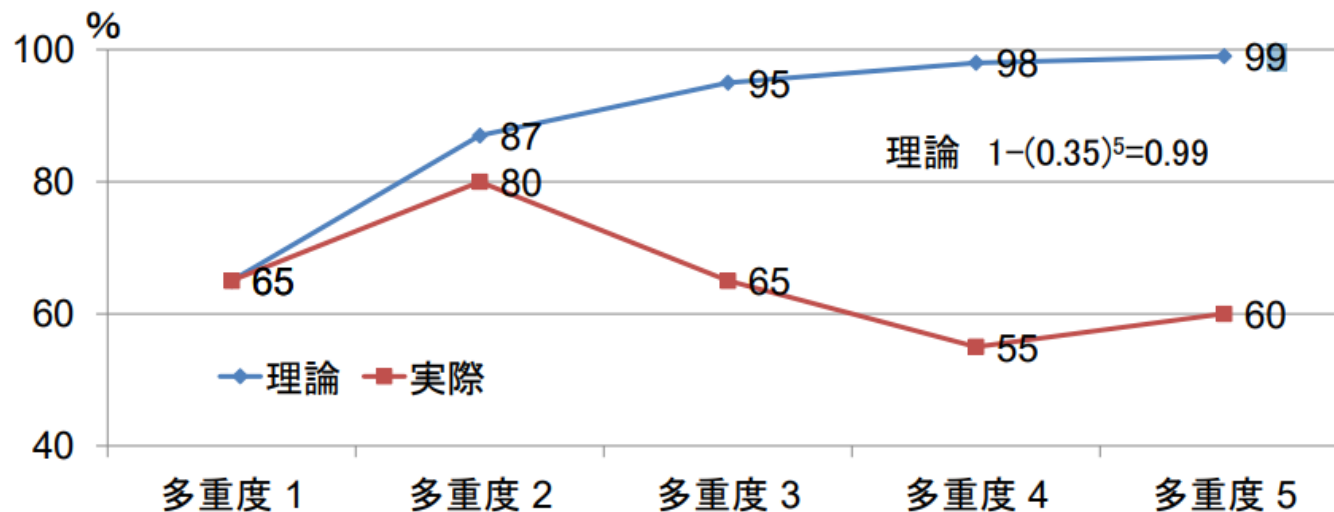


なぜレビュー観点が必要なのか？(4)

“観点”がない場合、メンバー間での重複確認や誰も見ていない抜け・漏れが発生しやすい



ぼんやりレビューすると、みな上から順に読んで書かれていることに反応して指摘する
→同じ指摘が重複する（重複チェック）



確認の多重化とエラー検出率
島倉大輔・田中健次(2003)品質33: 104-112

三重チェック以降は効果がない
(誰かが見るから自分はいいや 等)

【実験】Adhocレビューだとどの程度重複する？

1	3	3-8	ビジネス目的、システム化目標など、上位の概念や目標、計画などの確認	上位の根拠と矛盾がないか、足りないことはないか、そもそも何を達成したいのか、何を参考にこの要集仕様書を作ったか、何か似たようなバグれるものはないのか	目的・価値	企画書案要求仕様書	うれしの	
2	4	5	想定ユーザー、特にアウトプット利用者が困っていること、無駄だと思っていることの確認	経理や承認者、申請者が最近困っていることや非効率なことが解決されるのか	目的・価値	企画書案要求仕様書	うれしの	3
3	1	5	効率的にとは具体的に何を意図しているか、どのくらいの効率を担いたいのか	不明点の明確化	よしざわ			
4	1	5	システム化の目的には「社内の従業員が交通費を効率的に精算できる環境を提供する。」とあるが、すべて手入力であるなど効率化への配慮を欠いたシステムになっている	システム目的の実現性	あだち			
5	5	6	交通費の精算プロセスを自動化し、ヒューマンエラーを削減する。	どのようなヒューマンエラーを削減したいのか？	目的・価値	企画書案要求仕様書	うれしの	3
6	2	6	ヒューマンエラーとはどのようなことを意図しているのか？	不明点の明確化	よしざわ			
7	2	6	システム化の目的には「交通費の精算プロセスを自動化し、ヒューマンエラーを削減する。」とあるが、エラーチェック機能もなくヒューマンエラーへの配慮を欠いたシステムになっている	システム目的の実現性	あだち			
8	6	7	紙の使用を最小限に抑え、経費削減とSDGs対策を推進する。	・経費削減はどれくらい？ ・SDGsを社内目標などに落とし込むものがあるのか？何を推進したいのか？	目的・価値	企画書案要求仕様書	うれしの	3
9	7	7	紙の使用を最小限に抑え、経費削減とSDGs対策を推進する。	SDGs対策を「SDGsを意図に書き、〇〇を」「SDGs達成に向けた社内目標に従い、〇〇を」（SDGsは目標なり基準なりなので、続く言葉は「対策」ではおかしい）今までの何を次からどうしたいのか？	試用	企画書案要求仕様書	うれしの	
10	3	7	システム化の目的には「紙の使用を最小限に抑え、経費削減とSDGs対策を推進する。」とあるが、結果を印刷することが前提となっている。	システム目的の実現性	あだち			
11	8	8	精算プロセスの透明性	今までの何を次からどうしたいのか？	目的・価値	企画書案要求仕様書	うれしの	3
12	9	8	監査可能性	今までの何を次からどうしたいのか？	目的・価値	企画書案要求仕様書	うれしの	
13	4	8	システム化の目的には「精算プロセスの透明性と監査可能性を向上させる。」とあるが、監査ログ取得などの運用関連要件が不明のままになっている。	システム目的の実現性	あだち			

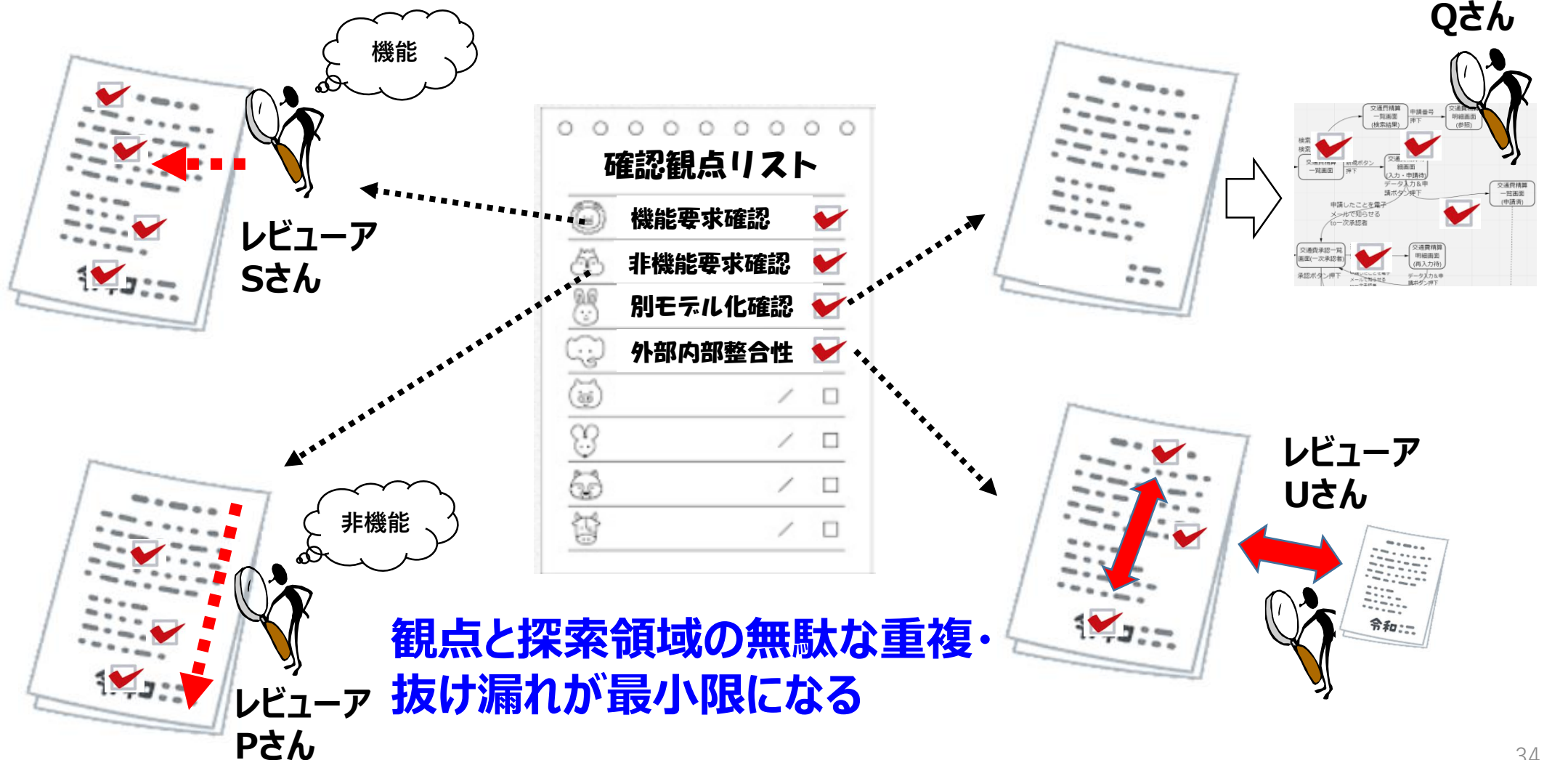
メンバー4名で、とあるレビュー対象をそれぞれレビューし、記録した。

- ・指摘総数：のべ92件（52種類）
- ・重複なし指摘数：25種
- ・重複した指摘：27種類
- 複数人が指摘した件数：計67件

同一指摘（ムダ）：67 - 27 = 40件分

なぜレビュー観点が必要なのか？(5)

“観点”があれば、メンバー間で確認する観点の分担が可能になる



なぜレビュー観点が必要なのか？(6)

有効指摘のノウハウを再利用しやすくなる（有識者依存度を低減できる）

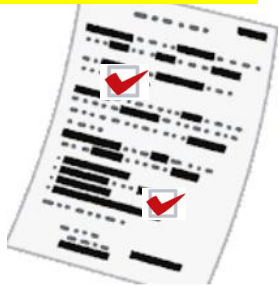
ABC Project



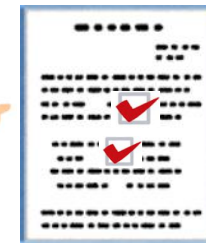
XYZ Project



RFV Project



YHN Project



取捨
選択

取捨
選択

取捨
選択

取捨
選択

有効なレビュー観点をまとめて再利用することでレビューの効果を高める

適切な思考方法＝簡単にマネできない、他者と差別化できる重要な技能 ただし・・・

単発・形式的な取り組みでは適切な実践ができない

- 観点設定→観点に基づくレビューを、**思考を伴わず形式的に行っても効果は期待できない。**
- 重要なのはレビュー対象や観点設計への理解に基づく実践。
そこには**状況に応じた「適切な思考」**と**「モデル化」**が求められる。
- その**実践力は、継続して取り組むことで身につくもの。**
当初は時間がかかる～失敗も多いが“失敗により”学習が進む
[実践→ふりかえり]の継続～時短＆効果向上



ふりかえり

実施内容に対してふりかえり、結果を残しましょう

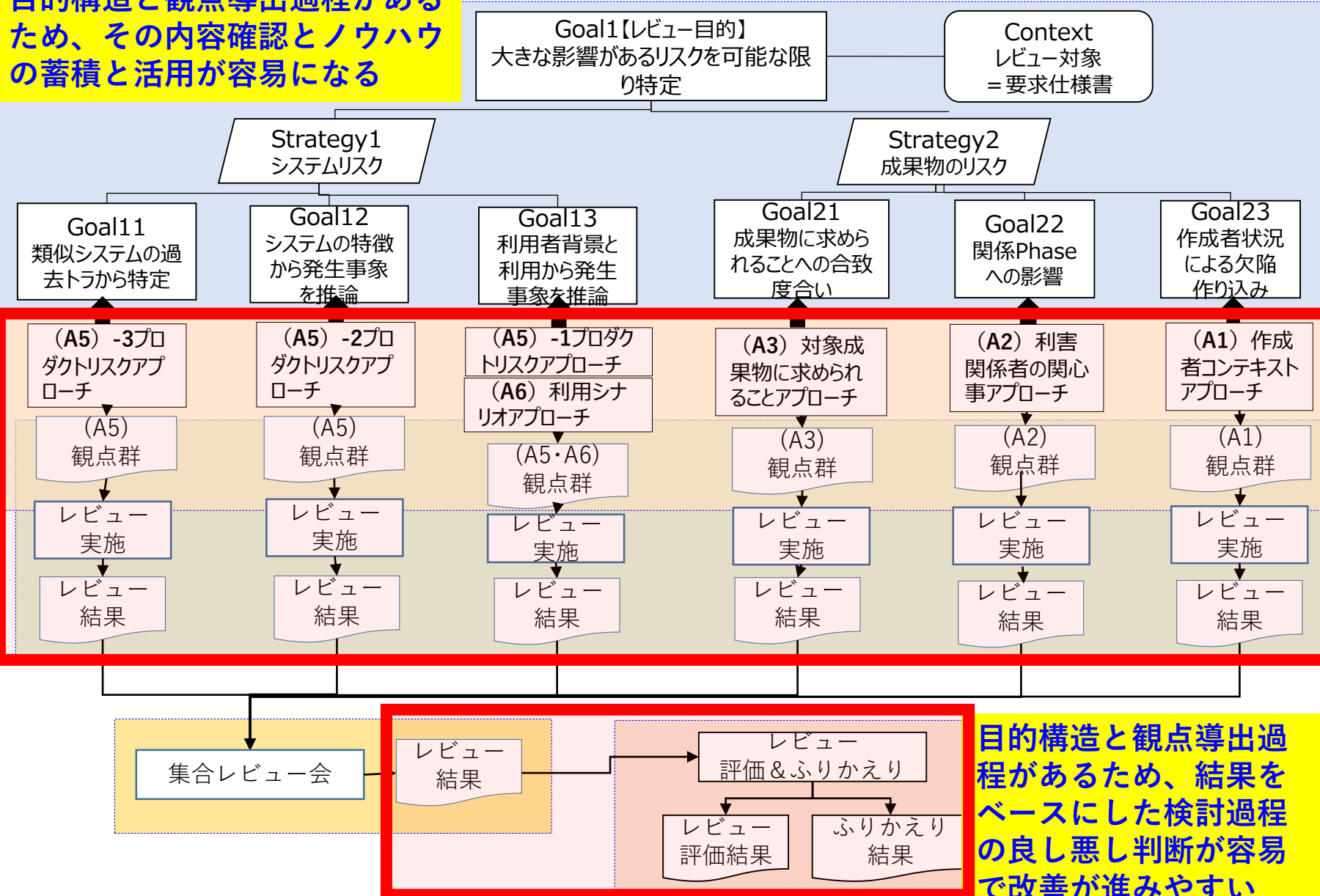
Keep:わかったこと・気づいたこと・感じたこと など	Try:今後の対応や方向性
Problem:現状の問題点や課題	

観点に基づくレビューワーク

レビューアプローチの全体像

ワーク実施領域

目的構造と観点導出過程があるため、その内容確認とノウハウの蓄積と活用が容易になる



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

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

ワークの全体像

前提条件 共有	レビューの背景・目的・対象・制約条件等の共有	3分間
ワークスタート		
Work1	レビュー目的達成のためのアプローチ概要設計	20分間
Work2	サブゴールへのレビュー観点導出手法割り付け	15分間
Work3	レビュー観点導出	25分間
休憩		
Work4	レビュー実施	15分間
Work5	レビュー結果サマリ・評価	20分間
Work6	チームふりかえり	10分間

解説中心

レビュー対象物

・ 交通費精算システム要求仕様（案）

P1.背景説明

1 A社 交通費精算システム要求仕様書

2

3 背景

- 4
- 5 A社で交通費精算を行う際の手順は下記の通り。
- 6 ・申請者は紙の交通費精算書を手書きで記入し、上長に申請する。
- 7 ・上長は申請内容を確認し、問題なければ押印し、経理部に送付する。
- 8 ・経理部では部長が申請内容を確認し、問題なければ押印し、会計担当者に渡す。
- 9 ・経理部の会計担当者は、経理部部長が押印した交通費精算書を参照しながら、
- 10 会計システムに入力する。
- 11 ・システム入力の3日後に社員への支払い（銀行振り込み）が行われる。

12

13 現行業務の問題点として、月初めの精算締めタイミングで、交通費精算の申請が集中することにより、会計担当者の会計システム入力の負荷が高くなっていることと、申請者が交通費精算書に手書きで記入するのが手間であるとの不満が多いことの2つが挙げられる。

14 上記の問題を解決するため、社内で交通費精算システムを構築することになった。

15

16

17

18

19 交通費精算システムでは、経理部でのシステム入力の手間を減らす対応と各個人の手書き記入の手間の解消を行う。交通費精算書の電子化については今回の対応には含まない。

20

21

P2.システム概要説明

22 1. 概要

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

- 38 交通費精算システムは下記の機能で構成される。
- 39 ・交通費精算一覧画面
- 40 ・交通費精算明細画面
- 41 ・交通費承認一覧画面
- 42 ・交通費商品明細画面（当資料では左記画面の説明は省略）
- 43 ・交通費精算書出力

44 交通費精算システム利用時は社内ポータルサイトから遷移することとし、社内ポータルサイトのメニューに交通費精算一覧画面と交通費承認一覧画面へリンクを追加する。このシステムへのログインはシングルサインオンとする。

45 利用ユーザは社内システムが利用できる全ユーザの想定である。

46 交通費精算システムで、精算データを申請すると、承認フローが実施される。承認フローの経路は既存のマスタを参照する。（承認フローは、申請者→一次承認者（所属部上長）→二次承認者（経理部部長）の順で実施される。）

47 交通費精算を申請した際に、一次承認者にメールで通知される。一次承認者が承認すると二次承認者にメールで通知される。二次承認者が承認すると承認フローが完了し、申請者にメールで通知される。

48 承認フローが完了すると交通費精算データは会計システムに自動連携される。

49

50

51

52

53

54

55

【交通費精算一覧画面イメージ】

交通費精算一覧				
従業員：P3.交通費精算一覧				
交通費一				
申請番号	申請日	精算額	申請者	申請状況
xxxxxxx	yyyy/mm/dd	¥9,999,999	〇〇太郎	□□□□□
15120001	2015/12/15	¥15,000	〇〇太郎	上長承認済
15121234	2015/12/28	¥1,200	〇〇太郎	申請済
(省略)		-	-	-

【交通費精算明細画面イメージ】

交通費明細						
申請者：P4.交通費精算明細						
申請番号：申請						
No.	利用日	行先	区間（発）	区間（着）	交通手段	往・片
99	yyyy/mm/dd	△△支社	〇〇駅	××駅	□□□□	▼▼
1	2015/12/10	横浜支社	東京駅	横浜駅	J R	往復
2	2015/12/11	川崎工場	東京駅	川崎駅	J R	往復
(省略)						

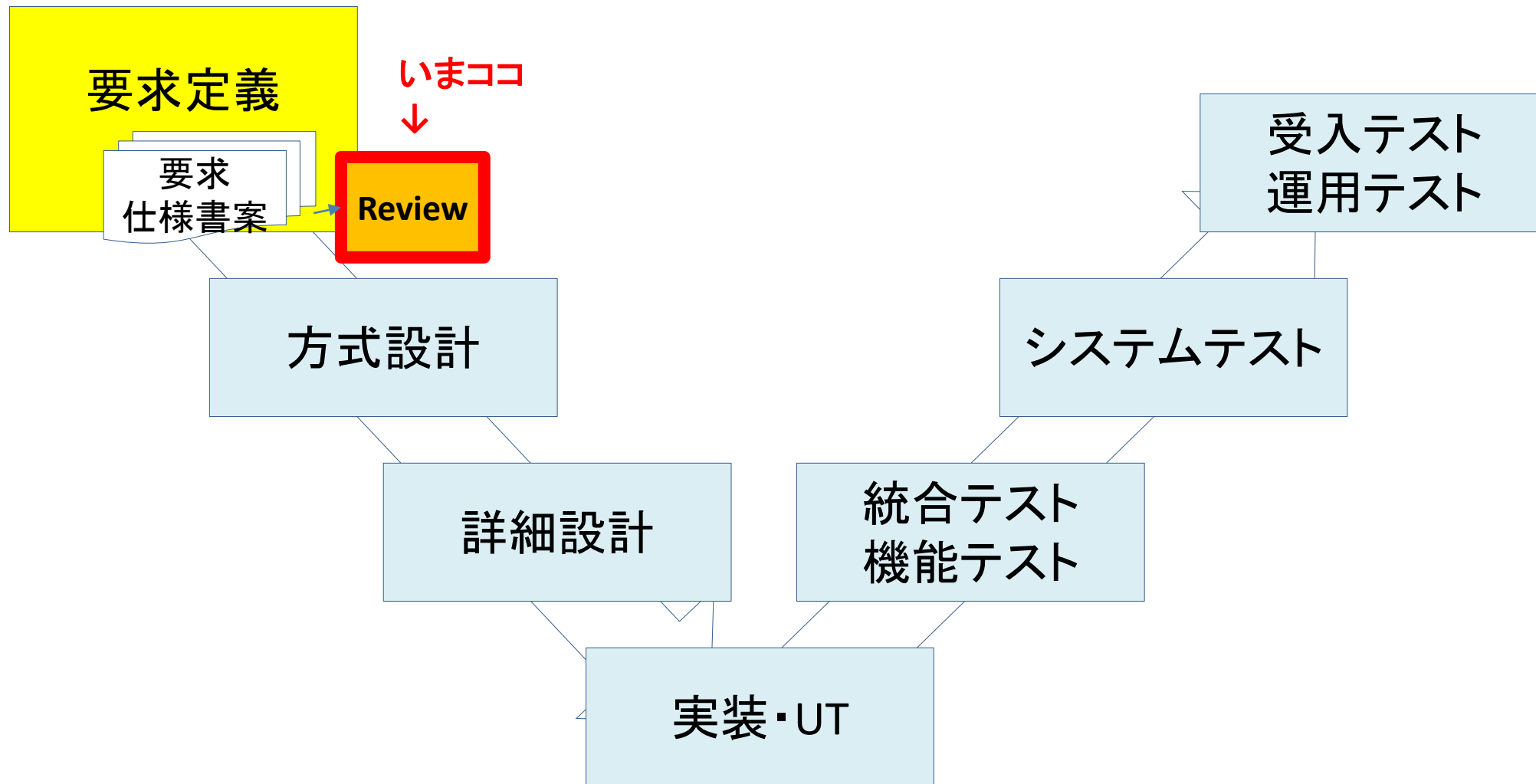
【交通費承認一覧画面のイメージ】

交通費承認一覧					
承認者：P5.交通費承認一覧					
申請状況承認					
交通費一					
選択	申請番号	申請日	精算額	申請者	申請状況
☒	Xxxxxxxx	yyyy/mm/dd	¥9,999,999	〇〇太郎	□□□□□
☐	15120001	2015/12/15	¥15,000	〇〇太郎	上長承認済
☐	15121234	2015/12/28	¥1,200	〇〇太郎	上長承認済
☐	(省略)		-	-	-

【交通費精算書出力イメージ】

交通費精算書						
経理部長 上長 申請者						
所属：○						
氏名：○						
申請日：P6.交通費精算書出力						
利用日	行先	区間（発）	区間（着）	交通手段	往・片	交通費
yyyy/mm/dd	〇〇支社	〇〇駅	××駅	□□□□	▼▼	¥9,999,999
2015/12/10	横浜支社	東京駅	横浜駅	J R	往復	¥940
2015/12/11	川崎工場	東京駅	川崎駅	J R	往復	¥620
2015/12/11	川崎工場	川崎駅	川崎工場	タクシー	片道	¥2200
2015/12/11	川崎工場	川崎工場	川崎駅	タクシー	片道	¥2350
(省略)						

システム開発フェーズと現在



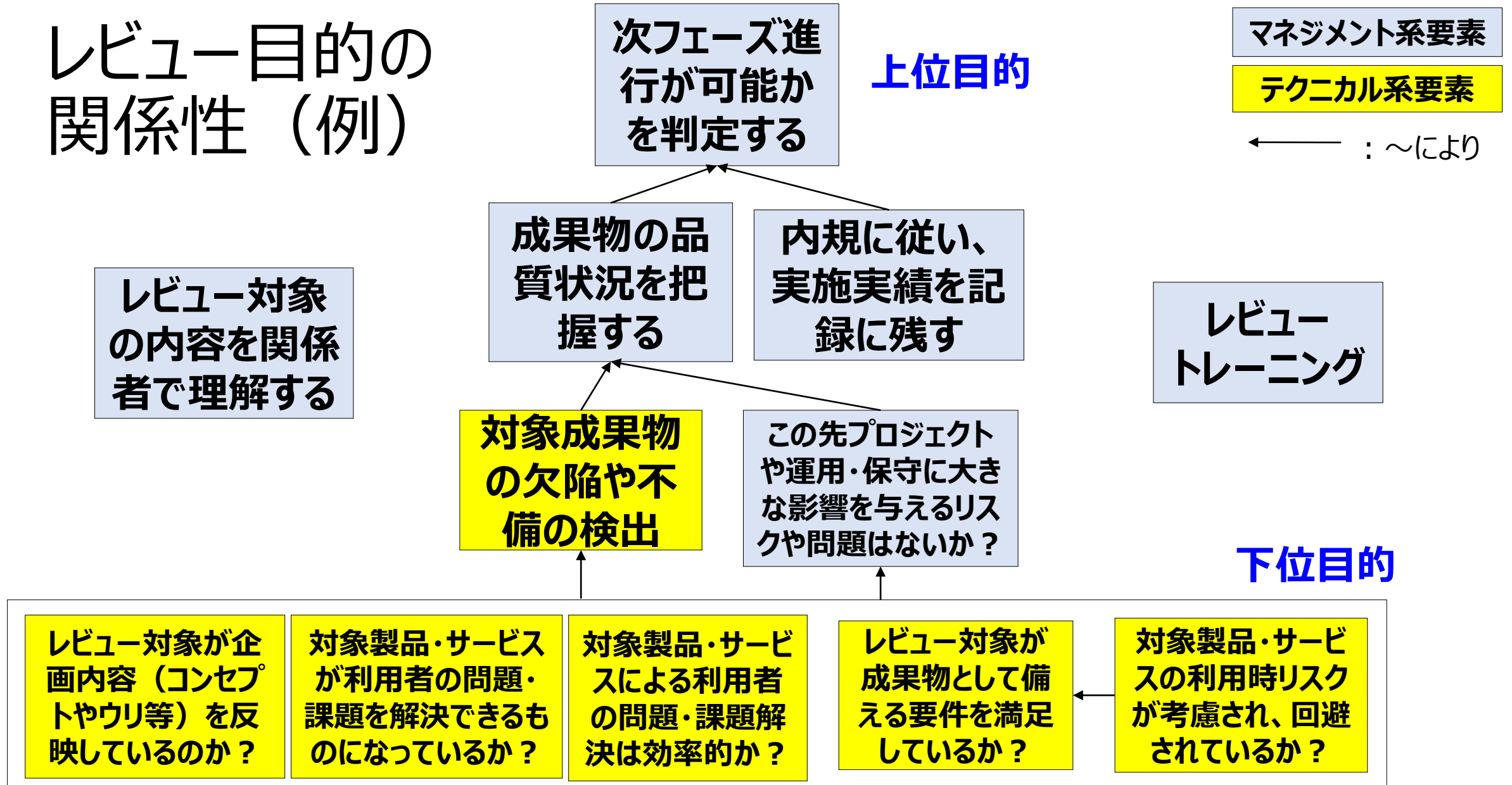
今回のレビューの背景、ニーズ、制約事項

- ☑対象成果物「〇〇要求仕様書」に対する初めてのレビュー。
- ☑対象物の記述構成、記述レベルなどは事前に把握済み。
- ☑レビューの準備に使える時間は1時間程度。
- ☑「レビュー実施」に使える時間は15分間。
- ☑人的リソース、時間が限られているので、当レビューを通じて大きな影響があるリスクを可能な限り特定したい。

※「リスク」とは“悪い影響を与える問題となる可能性”を指します。

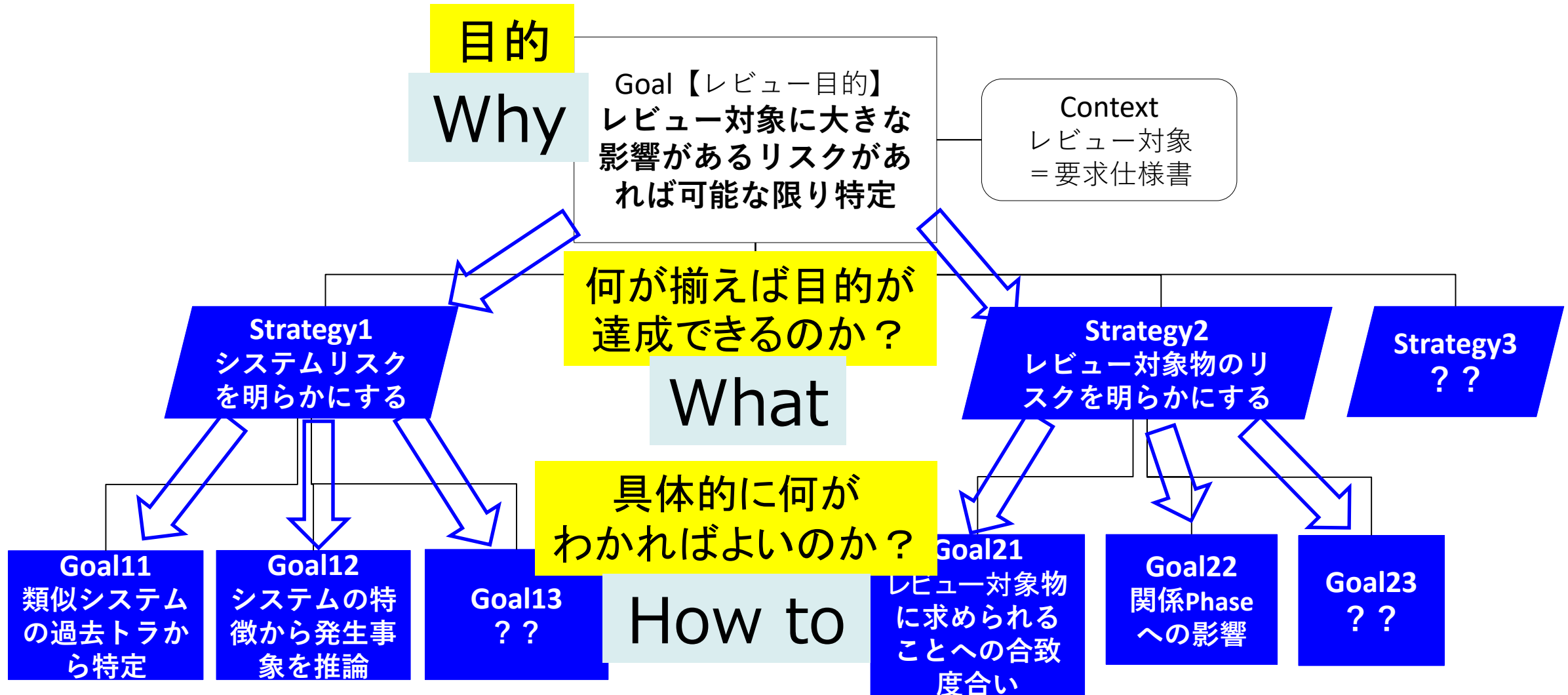
ワーク実践

レビュー目的の 関係性（例）

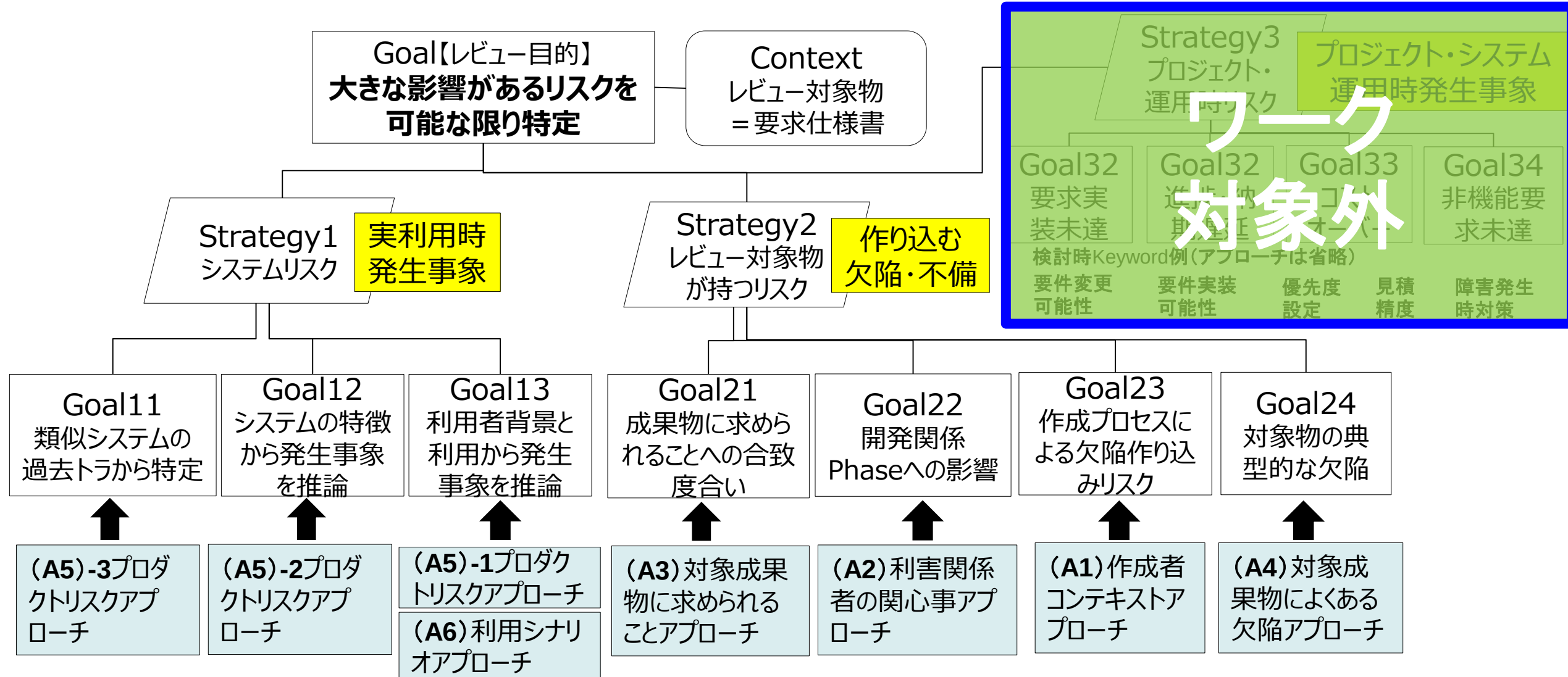


レビュー目的達成のためのアプローチ概要設計

目的(Goal1)→目的達成に必要な事項(Strategy)→必要事項の内訳(subGoal)



レビュー目的達成のためのアプローチ概要設計 と観点導出アプローチの割り当て[ワーク結果例]



用意されたレビュー観点導出アプローチ

- (A1)作成者コンテキストアプローチ
- (A2)利害関係者の関心事アプローチ
- (A3)対象成果物に求められることアプローチ
- (A4)対象成果物によくある欠陥アプローチ
- (A5)プロダクトリスクアプローチ
- (A6)利用シナリオアプローチ

※自ら所有する別アプローチを採用してもOK
→その場合はA7からの連番を利用



(A1)作成者コンテキストアプローチ



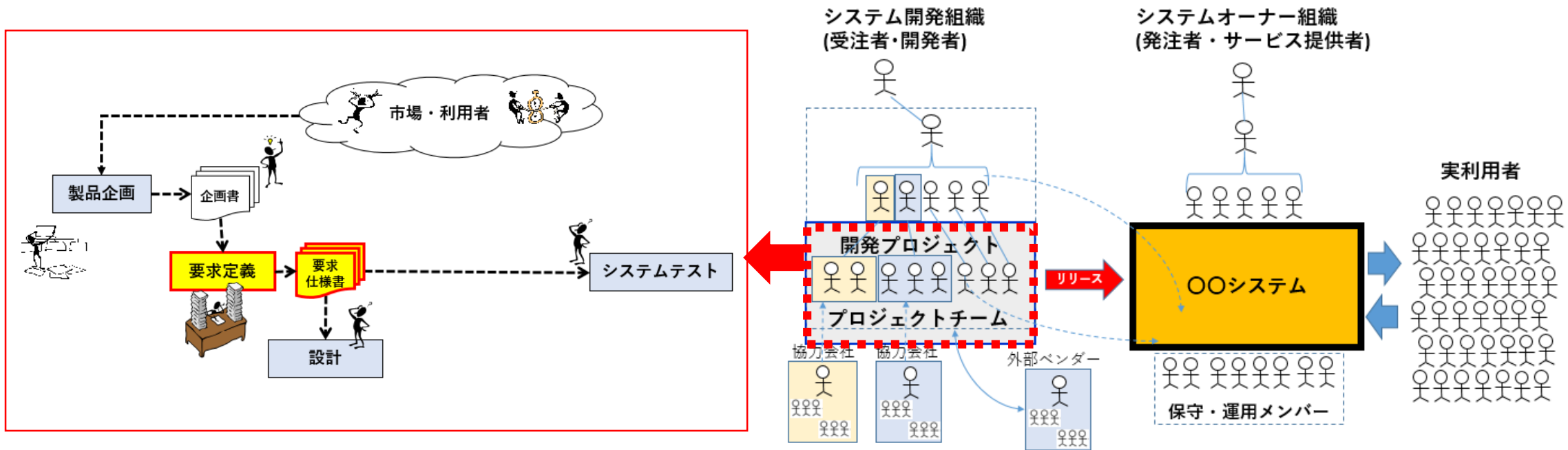
作成者のコンテキスト＝対象成果物がどのような状況の中で、どのように作成されたのかから、レビュー観点を導出する。

- (1)作成者はシステム開発経験3年目。これまでは設計のサブ担当者であったが、今回初めてメイン設計担当として割り当てられた。この分野のシステムを手掛けるのは初めて。
 - (2)レビュー対象成果物の作成に着手したのは2日前とのこと。
 - (3)作成者は、過去に参加したプロジェクトで構築したシステムの障害対応に2か月前から借り出され、継続して残業、休出が突出していた。障害対応は現在も継続中。
 - (4) レビュー対象成果物（MS-Wordで作成）の最終書き込み日時は今日のAM3:35。
- また、当初書き込み日時は、5年前の3/28 AM2:58。

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？

(A2)利害関係者の関心事アプローチ

図を参考に利害関係者を明確化→関心事→観点・対象を明確化



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？

(A3)対象成果物に求められることアプローチ

確認方法を明確化すること＝レビュー観点導出とする

ソフトウェア要求仕様の目次例

観点

1.はじめに

ドキュメント目的
記述範囲
用語定義
参考文献
全体構成

2.製品の背景と概要

製品の背景
製品機能
ユーザー特性
制約

要求項目の仮定と依存関係

3.具体的な要求事項

外部インタフェース

機能

性能要求

論理データベース要求

設計の制約

非機能要求等ソフトウェア特性

要求仕様の段落構成

確認方法

例.目次に該当する事項が存在しているかを目視で確認する。

例.解決したい利用者の課題に対して必要な機能が漏れなく明記されているかを目視で存在を確認する

ソフトウェア要求仕様が持つべき特性

観点

確認方法

正当性 (Correct)
システムに対するすべての要求が含まれ、以外の要求を含まないこと

無曖昧性 (Unambiguous)
全ての要求の意味が一意に識別されること

完全性 (Complete)
次をすべて含んでいること＝(1)すべての必要な要求、(2)すべての入力データと状況に関する応答の定義、(3)用語および図表の説明

一貫性 (Consistent)
要求間で矛盾がないこと

順位付け (Ranked for importance and/or stability)
要求が重要性や安定性に関して順位付けられていること

検証容易性 (Verifiable)
すべての要求に対して有限のコストで評価可能な手続きが存在し、検証できること

修正容易性 (Modifiable)
要求の変更に対して、容易かつ完全で一貫性を保って修正できるような構造を持つこと

追跡性 (Traceable)
要求の根拠が明確で、開発工程全体で参照できること

例：要求の背景や理由→要求＋制約条件が漏れなく追跡できるかを企画書と要求仕様で目視確認する

引用：要求工学：第3回要求仕様

<https://www.bcm.co.jp/site/2004/2004Dec/04-youkyuu-kougaku-12/04-youkyuu-kougaku-12.htm>



(A4)対象成果物によくある欠陥アプローチ

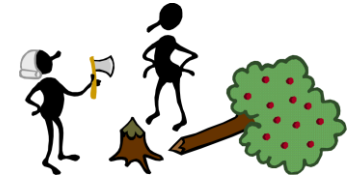
確認方法を明確化すること＝レビュー観点導出とする

観点	確認方法
☐ ユースケースや業務フロー、機能などの開始、終了条件の不備や条件によるアンマッチ	
☐ 信頼性、効率性、保守性などの非機能要求の不備や具体化不足（非現実的な網羅的設定の場合もある）	
☐ システムのSCOPEや他システムとの境界定義の不備	
☐ 要求間、機能間連携部分の例外事項などが一方の備考欄に記述され、他方で認識されない	
☐ 実現優先度未設定	例：要求事項単位に優先度が付与されているか＋判断根拠が把握できるかを目視確認する

引用/参考：「間違いだらけの設計レビュー改訂版」 森崎 修司著



(A5)プロダクトリスクアプローチ



1)利用者の立場、こんな使い方をすれば〇〇が発生するかも！を検討してみる。

2)システムの機能などから「こんなことが起きるかも」を想定してみる。

参考：安全リスクには健康リスク・経済リスク・環境リスクがある

3)類似製品の事故や障害、問題発生事例を確認してみる。（例：Web検索で探す）

どのような事象？	発生要因と影響	関係する機能	レビュー観点と対象
「湯沸かしポット」の例。 給湯時にやけどやケガをする	給湯速度が速すぎる→熱湯が 飛び散る→手をやけど→持って いた器を落として破損or足に落 として怪我	給湯機能	お湯の飛び跳ねを考慮した 給湯速度設定をしているか 対象：給湯コントロール

(A6)利用シナリオアプローチ

※利用シナリオ導出＝レビュー観点導出とする

利用者の典型的な課題を解決する利用シナリオ(通常+例外)を作成する。
それをベースに対象成果物をトレース（レビューを実施）することで、不明点や指摘事項があれば記録する。

<解決したい課題例>

業務で使用した交通費(往復分)を精算

往路：さっぽろ→(地下鉄：340円)→新さっぽろ(徒歩移動)新札幌→(バス：210円)→森林公園
(復路はこのルートを逆に辿る（同一金額）ため省略)

■利用シナリオ（通常）

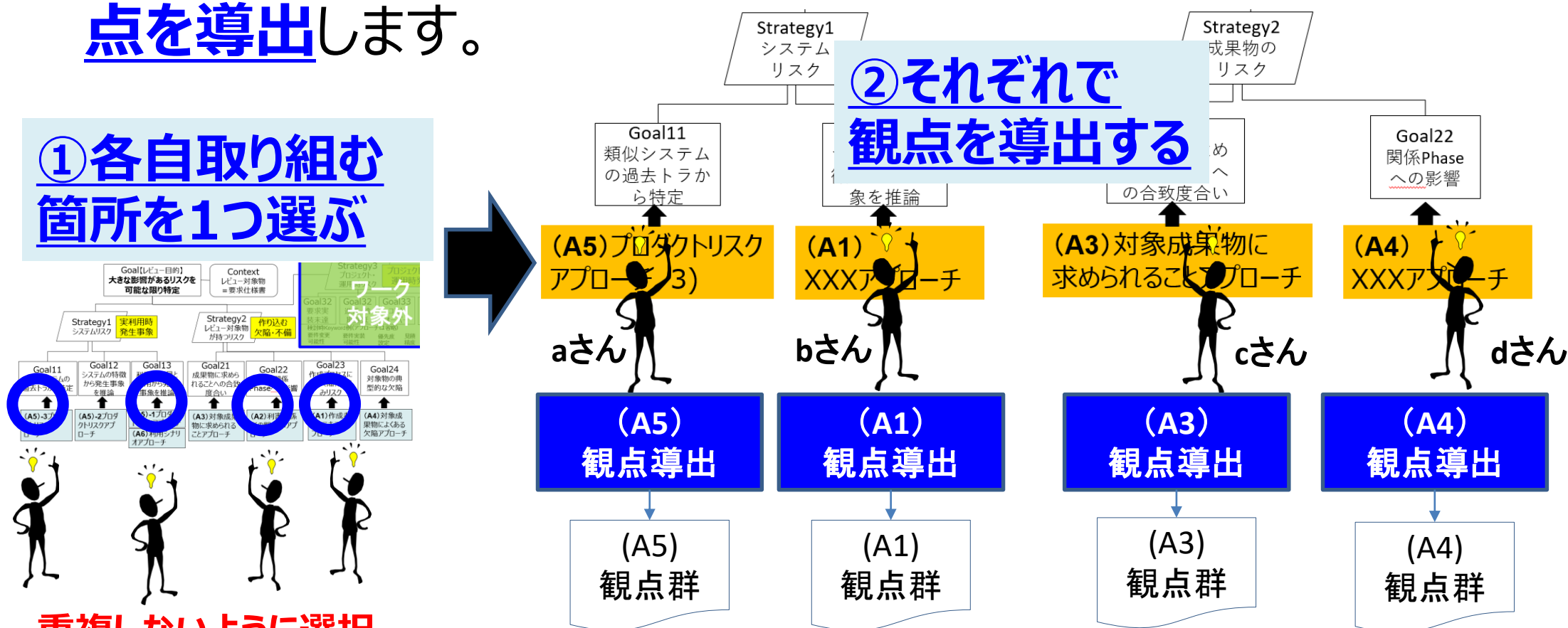
途中でエラーも発生せず、すんなり順調に入力・申請でき、承認後に口座に入金されるシナリオ。

■利用シナリオ（例外）

途中でさまざまなエラーに遭遇しながら入力・申請するが、結果的に否認されるシナリオ。
さらに否認後に再申請して承認され、最終的に口座に入金されるシナリオを付与するとよりよい。

レビュー観点導出

- チームで観点導出アプローチのうち各自選りすぐり1つを選び、観点を導出します。



観点設計がうまくできなかった事例

☑着眼点はよいが**観点導出に至れていないケース**

利用時に発生しそうな問題事象	発生要因と影響	関係する機能	レビュー観点
申請が期日に間に合わない	本人の体調不良等		

☑観点の**粒度が粗すぎる**ケース：レビューアが**その先の詳細化ができない**状態

着目した作業者の兆候	兆候からの仮説	想定される欠陥	レビュー観点
メイン設計担当が初体験	要求仕様書を作成するにあたっての知識が不足している	記載すべき事項が記載されていない	要求仕様書の内容として必要なものが十分に記載されているか？

☑観点の**詳細化に失敗**したケース

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)	レビュー観点
システムオーナー 保守・運用メンバー	当システムの保守担当	・保守・運用が容易か ・トラブル時に対応できるよう必要な情報がそろっているか	保守に必要な情報が取り揃っているか ～ドキュメントや協力会社体制など

ふりかえり

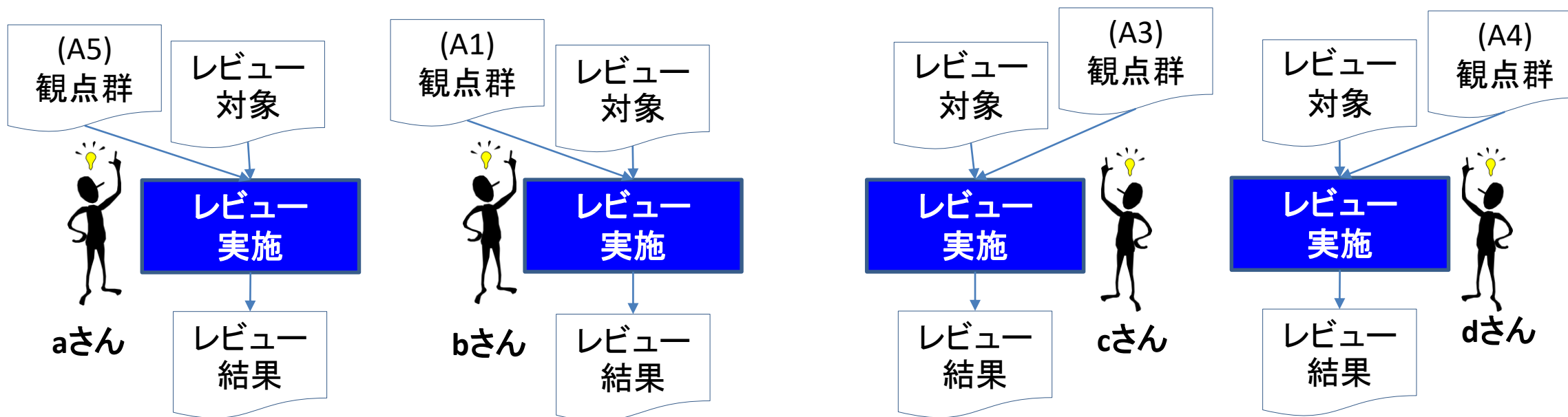
実施内容に対してふりかえり、結果を残しましょう

Keep:わかったこと・気づいたこと・感じたこと など	Try:今後の対応や方向性
Problem:現状の問題点や課題	

個別レビュー実施

- 各自で導出した観点に基づくレビューを実施します。

※観点に沿わない欠陥や不備を見つけた場合も迷わず記録する



レビュー結果記録様式

観点導出アプローチ番号： A			担当者：	
N o.	指摘箇所	指摘グレード 重・中・軽	指摘内容	観点
1				
2				
3				
4				
5				

観点に基づくレビューのふりかえり

(1) 観点設定レビュー結果評価

計30分間

20分間

No.	指摘箇所	指摘グレード 重・中・軽	指摘内容
1			
2	aさん		
3	レビュー結果		
4			
5			

No.	指摘箇所	指摘グレード 重・中・軽	指摘内容
1			
2	bさん		
3	レビュー結果		
4			
5			

No.	指摘箇所	指摘グレード 重・中・軽	指摘内容
1			
2	cさん		
3	レビュー結果		
4			
5			

No.	指摘箇所	指摘グレード 重・中・軽	指摘内容
1			
2	dさん		
3	レビュー結果		
4			
5			



評価指標	観点設定レビュー結果			
	aさん (A5)	bさん (A1)	cさん (A3)	dさん (A4)
効果・影響度大 主対象： 安全性未考慮 キャパシティ要件欠落	No.1 No.3	No.2	No.1 No.4	No.1 No.2
効果・影響度中 主対象： 性能要件未考慮 機能不足→使いにくい	No.2	No.1	No.3	No.3
効果・影響度小 主対象： 誤字・脱字・衍字 部分的画面遷移なし 画面項目不足	No.4		No.2	
指摘件数計	4	2	3	3

観点に基づくレビューのふりかえり

(2) 評価結果考察とふりかえり

計30分間

10分間



いつものレ
ビューと〇〇が
違うと感じた



欠陥指摘が◇◇
側に集まっ
てる！

評価項目	aさん(A5)	bさん(A1)	cさん(A3)	dさん(A4)
効果・影響度大 主目的： 安全確保 セキュリティ要件充足 運用負荷低減・保守性向上	2 3	6	9	11 12
効果・影響度中 主目的： 性能向上 信頼性向上 運用負荷低減・保守性向上	1	5	8	10
効果・影響度小 主目的： 開発・テスト・デモ 部分の負荷軽減 開発効率向上	4		7	
指摘件数計	4	2	3	3



結果はよいけど
理解できてない
ところがある

△△部分をこう
やって適用すれ
ばよいかも！

考察結果
事実
+ 結論or推論

気づいたこと
／感じたこと

疑問・不安
など

いつもの方法と
の違い／適用へ
のアイデア

その他

実務での活用
方法

ワーク結果解説

事前レビュー結果

アドホックレビュー結果の典型的な特徴（例）

☑成果物の前半部分での指摘数が多く、後半部分ほど指摘数が少なくなる。

→理由：先頭から読んで指摘することで、時間切れとなり／
または集中力低下により後ろ側に未確認領域が存在する。

☑記述内容に対する指摘が多く、記述がないことに対する指摘が少ない。

☑複数レビューアが同じ確認をし、同じ指摘が提示される。
→探索領域・観点→指摘の重複

当アプローチによる典型的な結果

△改善前 (Before) ●改善後 (After)		発見可能Phase (想定)				計
		実装・UT	IT	ST・OT	C/O後	
		1	3	5	7	
検出効果	効果大 主対象: 要件抜け・誤り	5 				30 ↓ 335
	効果中 主対象: 機能上のバグ (誤植による)	3 				75 ↓ 93
	効果小 主対象: 誤字・脱字・衍字 規約違反	1 				16 ↓ 8
計		35→19	18→0	40→340	28→77	121→436

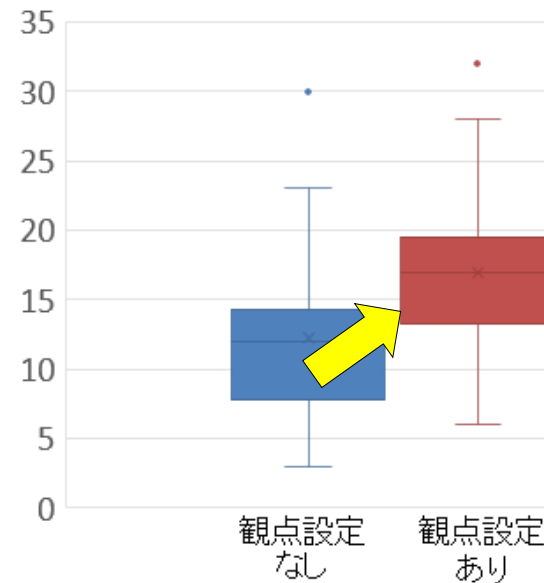
レビュー観点設計による典型的な指摘変化の例

△=アドホックレビュー指摘

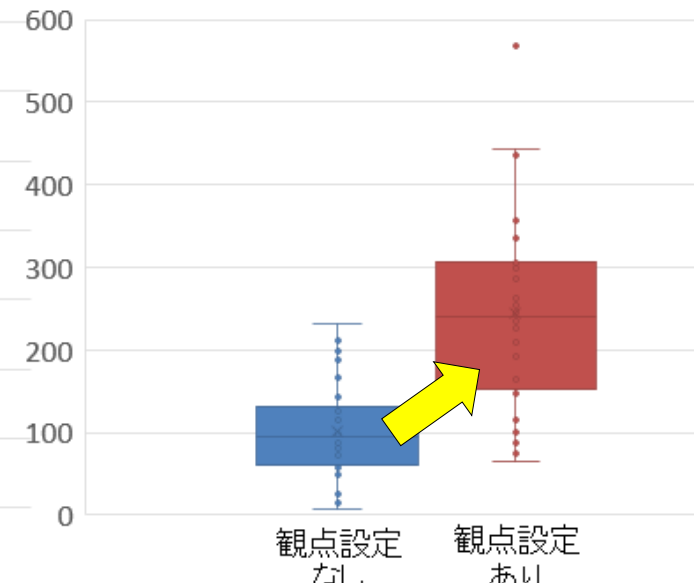
●=観点設計レビュー指摘

2012～2020年に実施した全ワークショップ°
(11社30チーム) 実績から算出

<指摘件数>

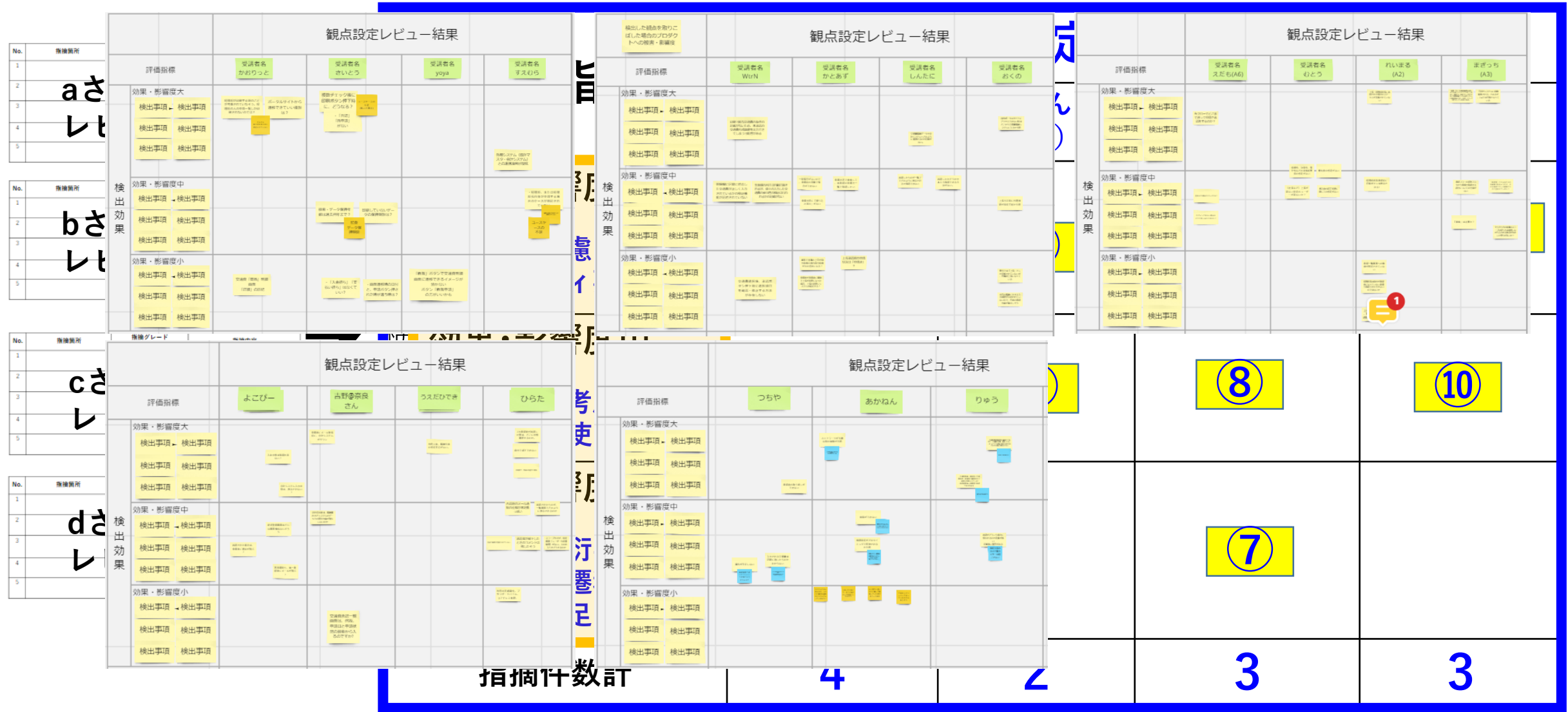


<指摘価値>



観点に基づくレビューの評価 (例)

レビュー結果の見える化
→ふりかえりに活用



指摘件数計

4

2

3

3

各チームのレビュー導出技法選択と指摘数

チーム	人数	レビュー観点導出技法選択と指摘数						指摘数計
		M1	M2	M3	M4	M5	M6	
A	4	3-2	6-2	1-0		2-1		12-5
B	4			1-1	9-3	3-1	5-2	18-6
C	4	4-1			3-1	4-1	6-0	17-3
D	3	6-5	3-1				3-1	12-7
E	4		5-1	6-2	4-2		3-1	18-6

※全指摘数(m件)とそのうちの効果大の指摘数(n件)を“**m-n形式**”で示した 計77-27

各チームのレビュー導出技法選択と指摘数

チーム	人数	レビュー観点導出技法選択と指摘数						指摘数計
		M1	M2	M3	M4	M5	M6	
A	4		3-1	3-0		2-0	3-2	11-3
B	4		3-0	2-1		3-1	4-0	12-2
C	4		1-0	1-1		1-1	8-2	11-4
D	4			1-0	1-1	2-0	7-2	11-3
E	4	1-0	0-0	4-2			2-1	7-3
F	3	4-1	3-1				8-3	15-5

※全指摘数(m件)とそのうちの効果大指摘数(n件)を“m-n形式”で示した 計67-20

ワークふりかえり結果

レビュー依頼

レビューイヤーのセルフチェックでも使える

レビュー対象を確認する事前準備の時間は大事

深い知識なくてもある程度のレビューできそう
見てほしい所を伝えておくとかで

レビュー依頼する時に、「XXの観点からレビューお願いします」みたいなコミュニケーションするとレビューの負担も下がるし精度も部分的にあがるかもしれない

今回のような仕様レビューに限らず、すべての「レビュー」と言える活動すべてで事前に観点を定めたり宣言したりするのは有用そう。少なくとも宣言することは明日から取り組みたい。

レビュー分析・設計・計画

レビュー導出アプローチをテラリングして各プロジェクトに適するものにして使用するとよいかと思いました。

抽象、具体と多角的にアプローチできた組み合わせるとパワフル

A2
自分と違う意見や背景からアイデアが出せた

色んな人とレビュー観点話し合えるって新鮮

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す。

誰がどのスキル・レビュー観点にふさわしいか把握する必要がある

チェックリスト作成がゴールでないため、目的に応じて選択できるようにしたい

レビューの設計はよさそう。使いまわしていきたい

レビュー観点導出のレビューをしなくなる

レビュー実施

過去の知見からのレビューになっていたが、レビュー観点導出で行うとまた違う観点で洗い出せるので良いと思った

A2
はユーザーの使い勝手などの観点からレビューできた

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

観点によって発見できる問題がぜんぜん違う

レビュー評価・ふりかえり

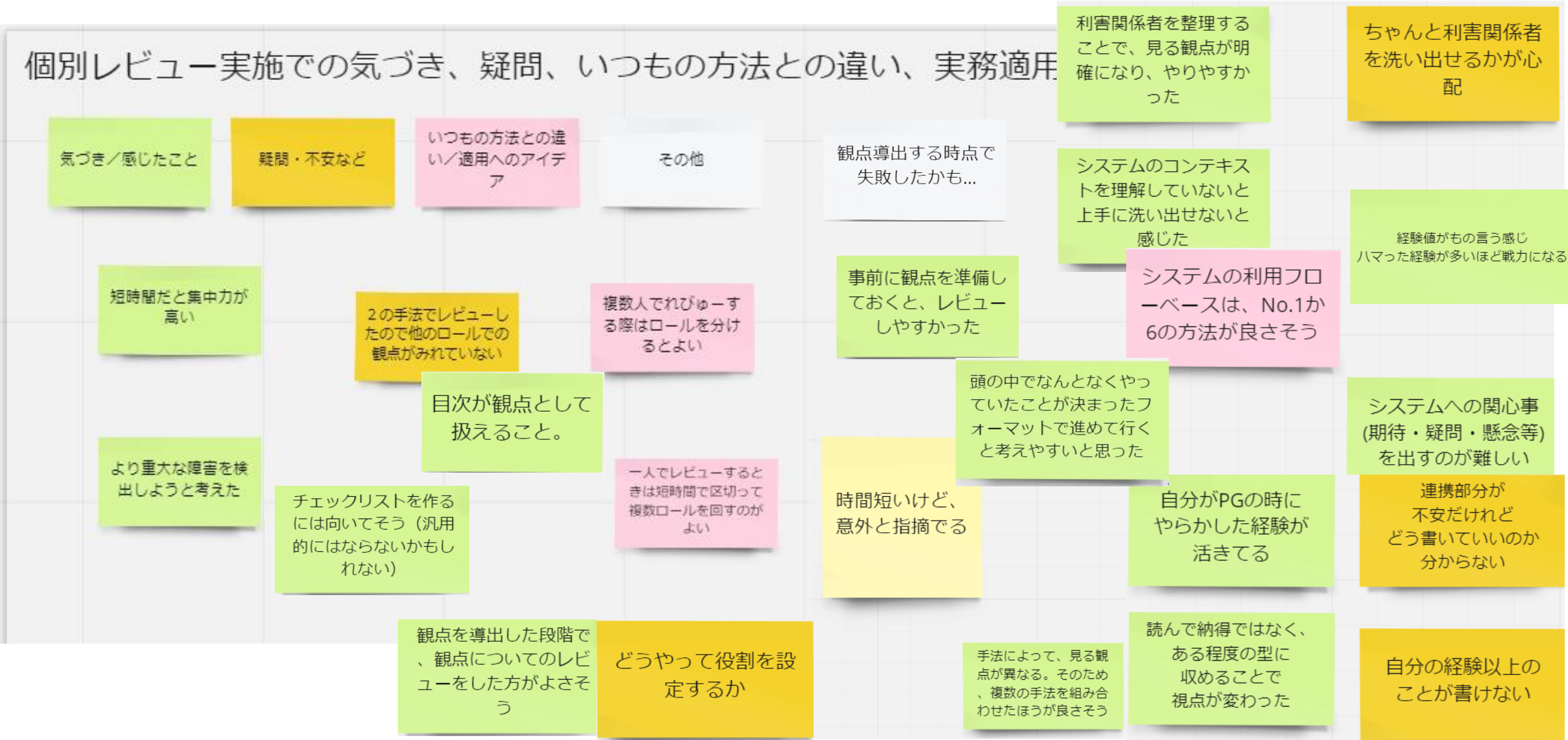
「良い観点」についての共通認識を取るのは難しそう。日々のレビューを定期的に振り返って、良いレビュー観点を育てていきたい。

レビューを振り返るプロセス、考えは今までなかった

派生開発で観点の使い回しできそうなので、観点を資産化する

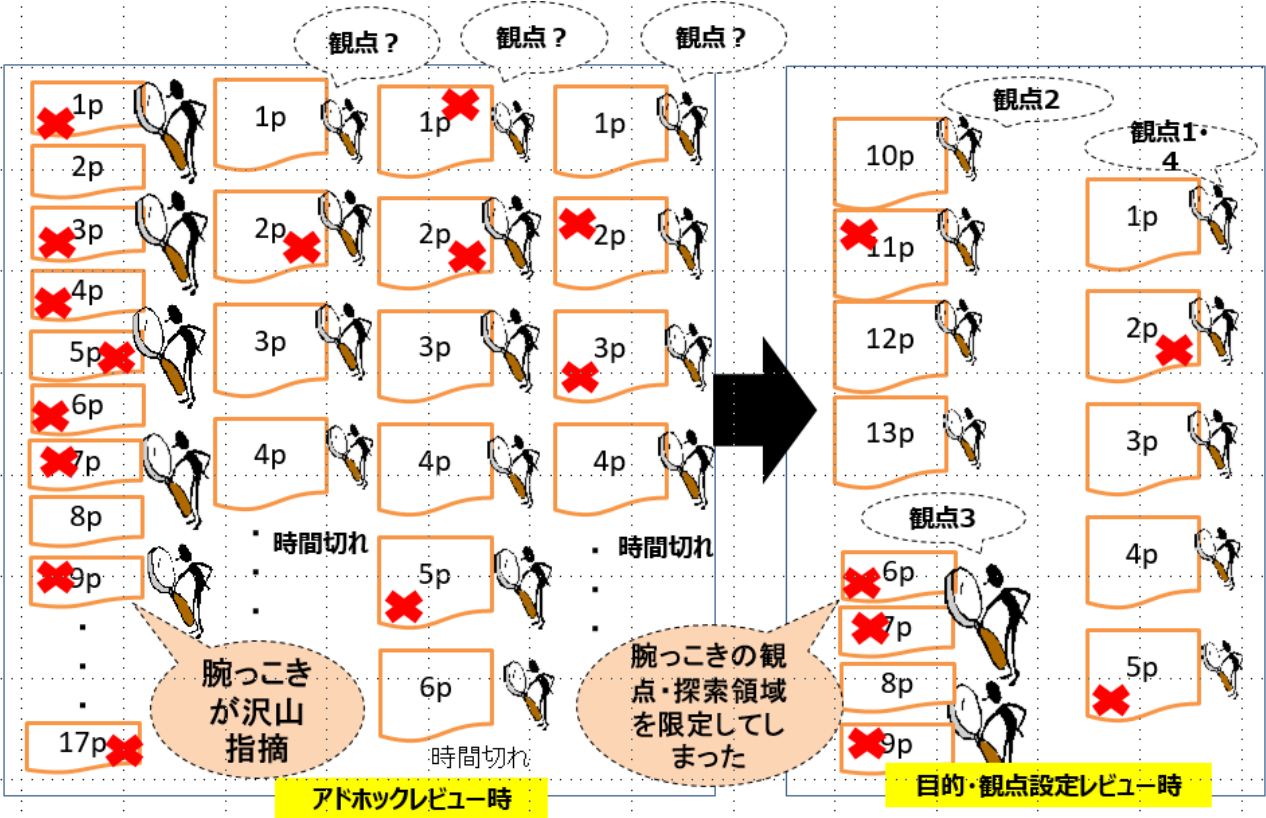
体系的なレビュースキルの蓄積、向上につなげられそう

ワークふりかえり結果



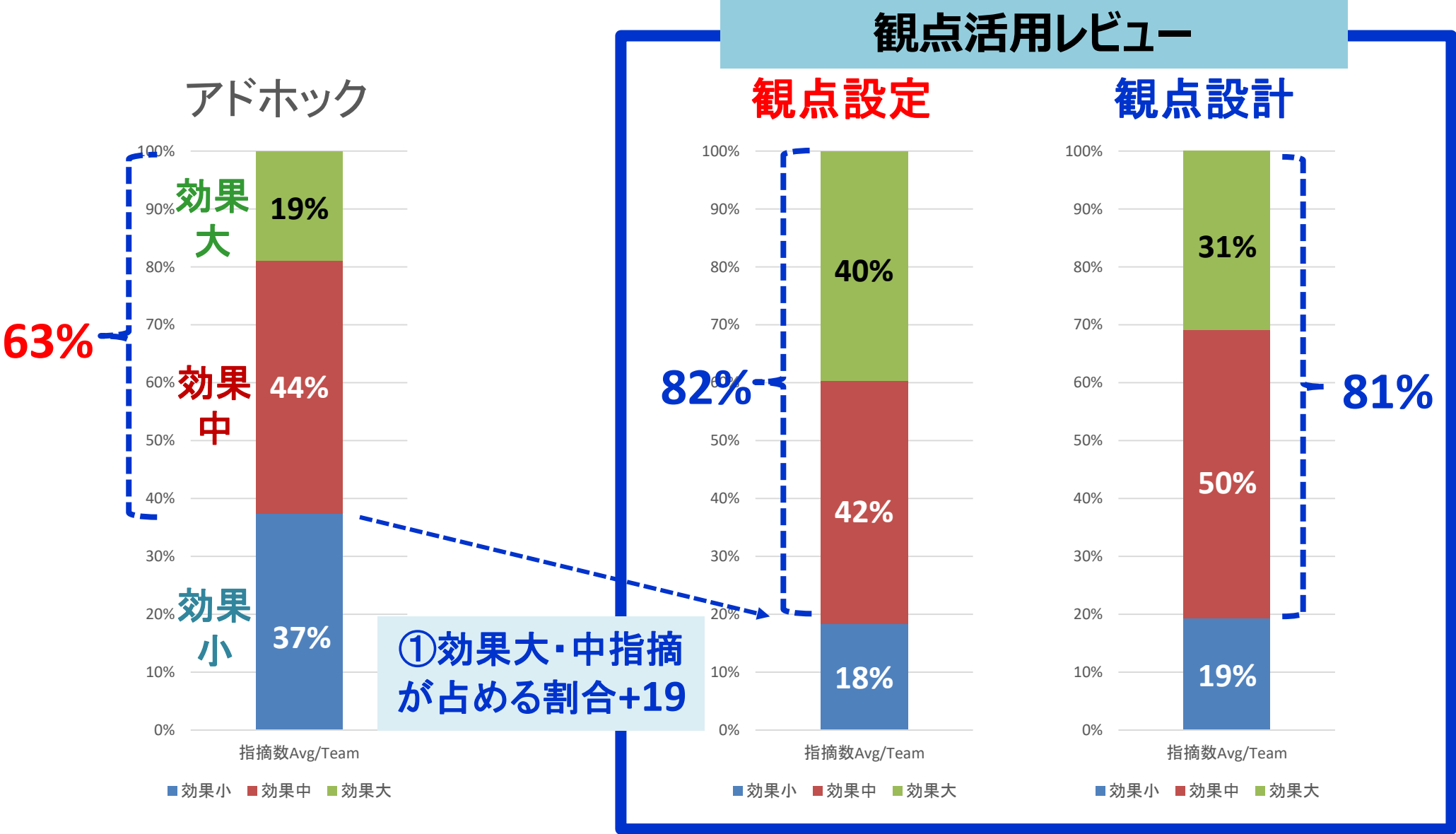
当手法でも効果が獲得できなかった事例

指摘件数: $\triangle 14 \rightarrow \bullet 11$



$\triangle$ 改善前 (Before) $\bullet$ 改善後 (After)		レビュー結果	計
検出効果	効果大 ハード変更必要 計測器故障	5	 0 $\downarrow$ 40
	効果中 沸点記述 整合性 デフォルト エラー後復帰×	3	 69 $\downarrow$ 18
	効果小 誤記 仕向け表記なし ロック押下仕様 長押し仕様	1	 9 $\downarrow$ 7
計		78 $\rightarrow$ 65	

観点活用レビューは効果大・中指摘増に寄与できる



今日うまくいかなかったり時間がかかったとしても 何度も実践して体得する類のノウハウなので要注意

- 以下の推論はおそらく誤りです。

**今日うまくいかなかった→このあともずっとうまくいかない
すごく時間がかかった→このあとも同じく時間がかかる**

- たしかに簡単にできるようになる類のノウハウではありません。
- しかし、**実践→失敗→見直しを繰り返すことで体得できます。**
- 体得すると・・・短い時間でずばりと適切な解を出せるようになります。

※これを時間をかけて経験的にできるようになった方 = “有識者”

ふりかえり

実施内容に対してふりかえり、結果を残しましょう

Keep:わかったこと・気づいたこと・感じたこと など	Try:今後の対応や方向性
Problem:現状の問題点や課題	

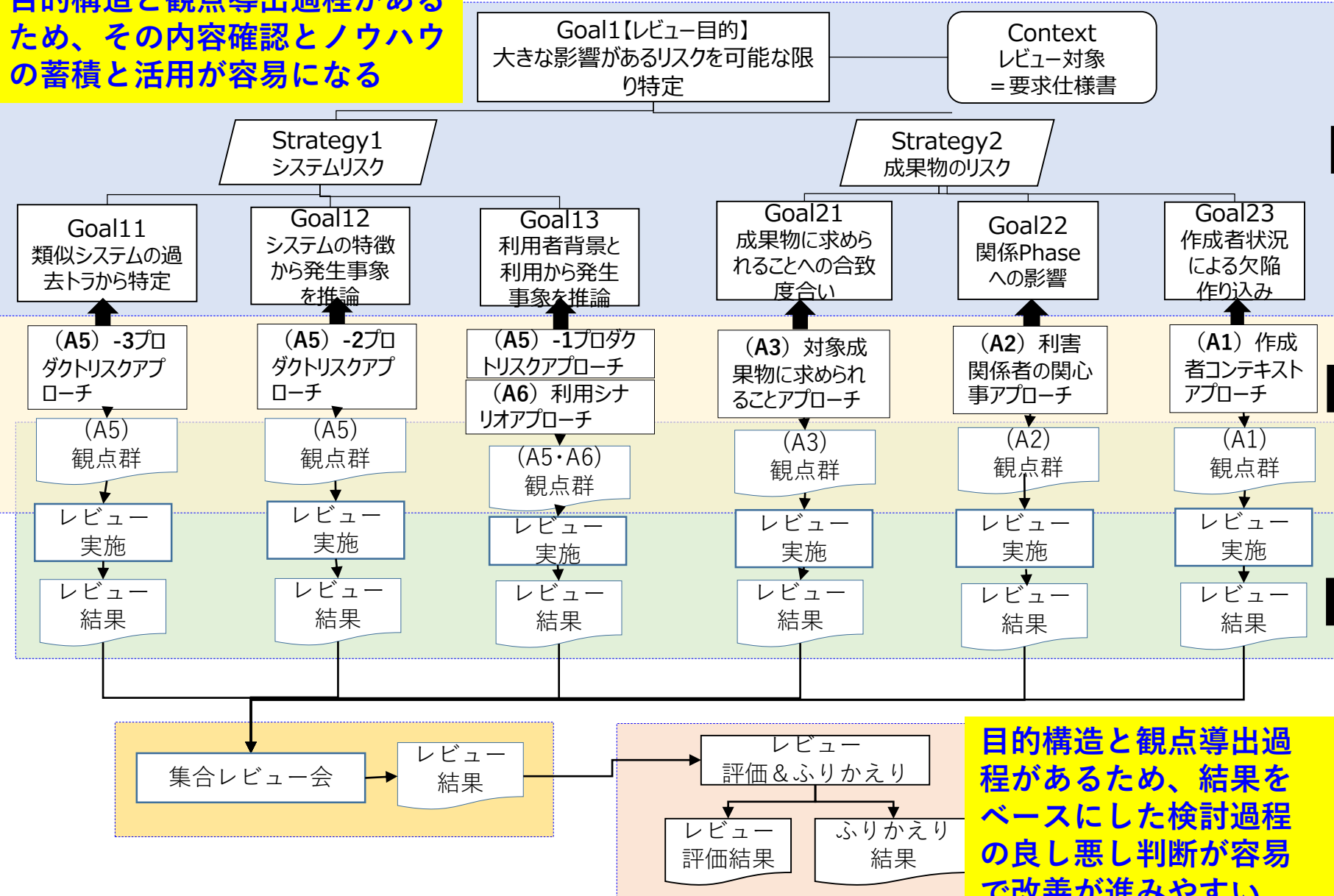
アプローチ解説

今回のワークの意図／提案する実践方法

よくあるレビューの状態	想定される結果	今回提案するレビュー実践方法
集合会議の際に初めて成果物を見ながら思い付きでレビューを行う。 ＝全員類似の観点で成果物を先頭から読んで指摘する。	☐ レビュー結果がレビューア的能力に依存する。 ☐ 同じ箇所を複数人で見るため指摘が重複するなど費用対効果が低くなりがち。 ☐ 成果物の後半がスカスカの結果になることも。	☐ 事前に対象を配布し、概要把握して観点設定＋観点割当て。 ☐ 個別レビュー実践、終了後に集合し、読み手による解釈共有・議論・結論と進める。
レビューの成果を有識者の参加に頼ってしまう。	☐ 非有識者は何もできないと諦め、いつも通りのレビューで終える。（あとは誰か見るだろう） ☐ 有識者の都合でレビュー待ちになる。（進捗遅延などに繋がる）	☐ 同上 ☐ 適切なレビュー技法を選択して活用する。
大量(页数)成果物を一度のレビューですべて見る。	☐ 長時間となるため成果物の後半は集中力が落ち、薄い確認になりがち。見落としが多くなる。 ☐ 成果物の質が悪いと、修復できずに通過してしまう可能性大。	☐ 意図的に対象と観点を分割して実施する。 ☐ 1回あたり30分～45分で実施する。 ☐ 作成過程で随時レビューを実施する。
対象物だけを見てレビューを行う。	☐ 記述されていることだけに反応する＝書かれていないことに指摘できない。見逃しが増える。	☐ レビュー対象に関係情報を付与し、その内容・状態から兆候→仮説→欠陥→観点へ。 ☐ プロダクトリスクや関係者の関心事等から観点を設定し、確認する。

レビューアプローチの全体像

目的構造と観点導出過程があるため、その内容確認とノウハウの蓄積と活用が容易になる



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]

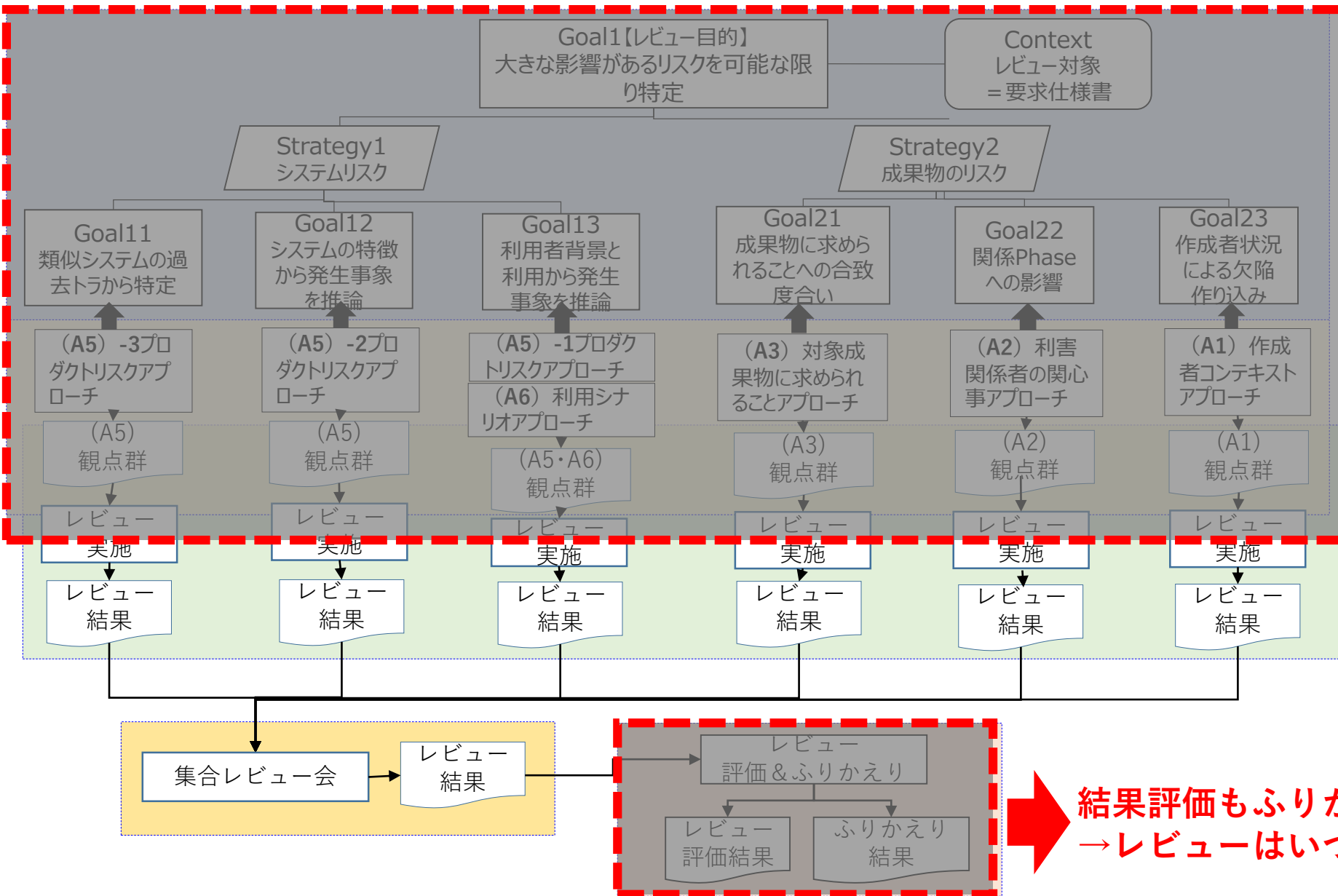
レビュー観点に基づくレビュー実施

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

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

よくあるレビューアプローチ



ここをすべてレビューの頭の中だけで暗黙的に実施する／あるいは形式的・固定的なチェックリスト等で済ませている

結果的に
レビューの成果が
有識者の参加有無に
頼る状態に

結果評価もふりかえりも行わない
→レビューはいつも同じやり方に

このアプローチの対象領域

**レビューで
確実に指摘をする
ためのノウハウ**

レビュー目的構造化～観点導出

レビュー会で指摘を
しやすくする場づくり

例：ファシリテーション

**業務知識によらない
レビューノウハウ**

業務知識による
レビューノウハウ

例1：物流系システム

例2：特定のITシステム

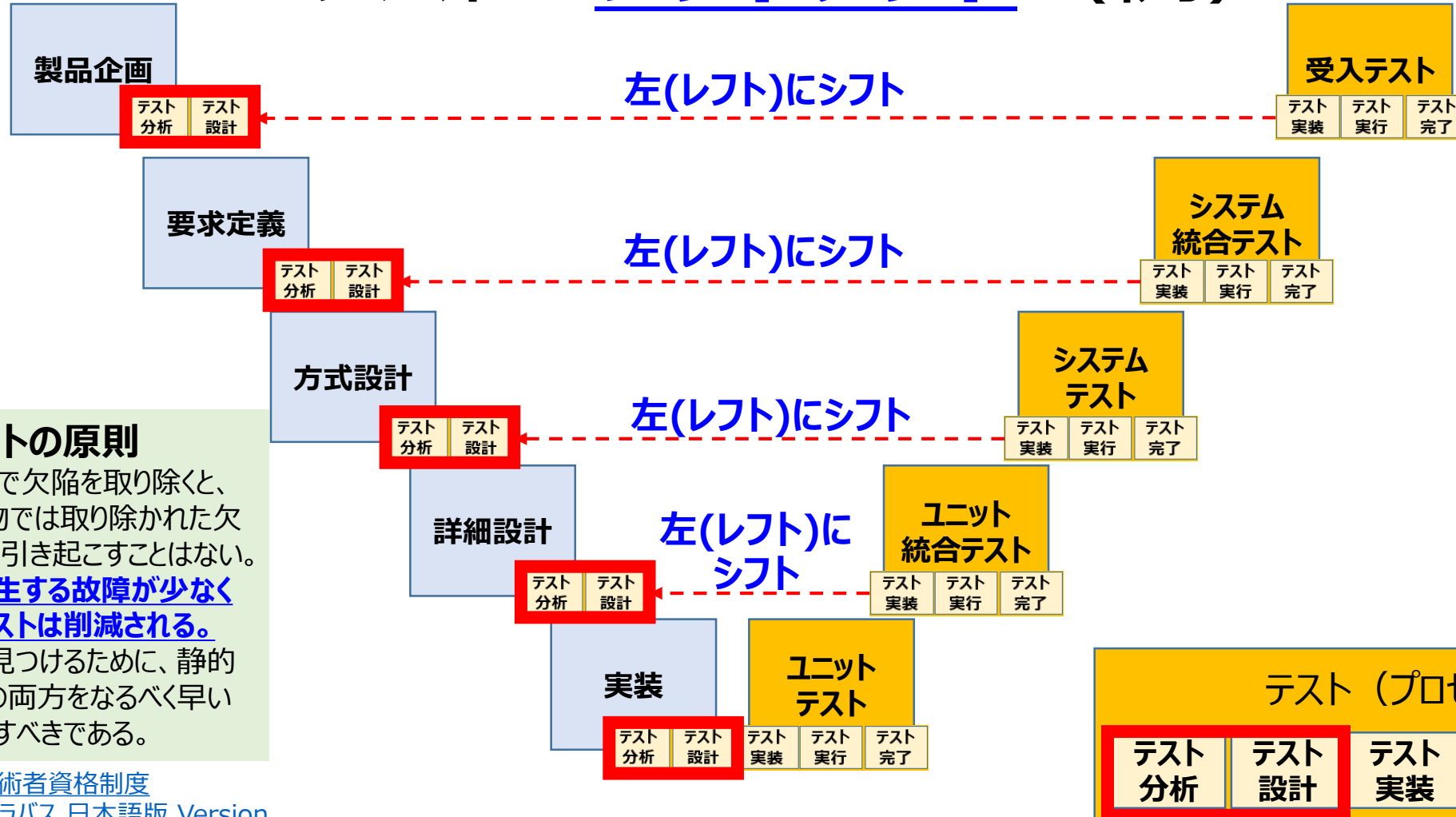
ふりかえり

実施内容に対してふりかえり、結果を残しましょう

Keep:わかったこと・気づいたこと・感じたこと など	Try:今後の対応や方向性
Problem:現状の問題点や課題	

ソフトウェアライフサイクルとレビュー

テストプロセスの前段部分を先出しする ～テストのシフトレフト（例）



早期テストの原則

プロセスの早い段階で欠陥を取り除くと、その後の作業成果物では取り除かれた欠陥に起因する欠陥を引き起こすことはない。

SDLCの後半に発生する故障が少なくなるため、品質コストは削減される。

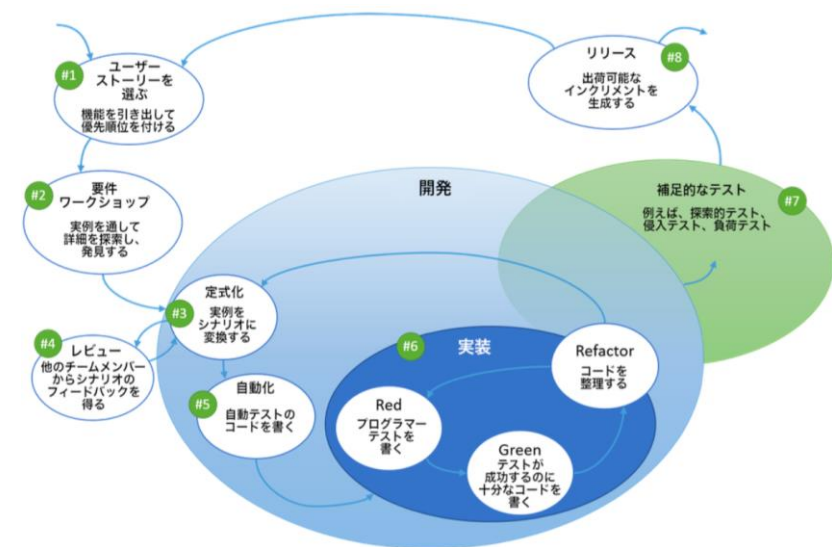
早い段階で欠陥を見つけるために、静的テストと動的テストの両方なるべく早い時期に開始すべきである。

引用：[ISTQBテスト技術者資格制度 Foundation Level シラバス 日本語版 Version 2023V4.0.J01](#)

シフトレフトの原理を活用した テスト主導ソフトウェア開発

BDD

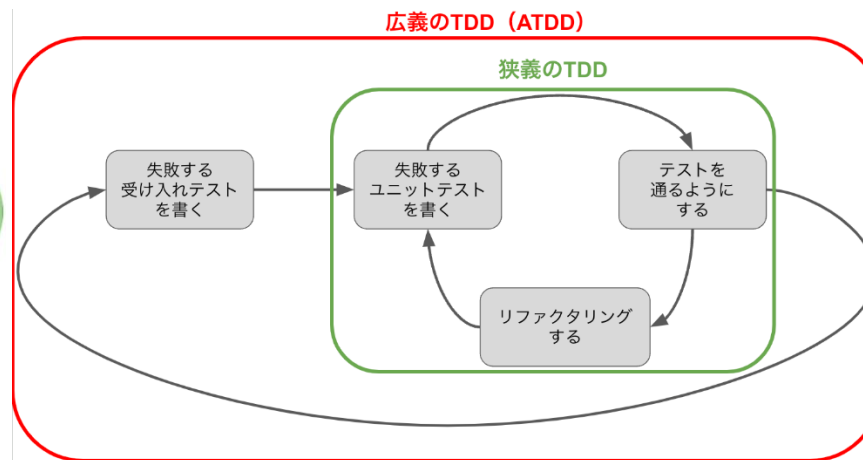
Behavior Driven Development
振る舞い駆動開発



TDDとBDD/ATDD(4)
ツールとしてのBDDとプロセスに
組み込まれたBDD より

ATDD

Acceptance test-driven
development
受け入れテスト駆動開発



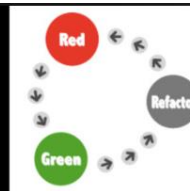
TDDとBDD/ATDD(3)
BDDとATDDとSbE より

TDD

Test-Driven Development
テスト駆動開発

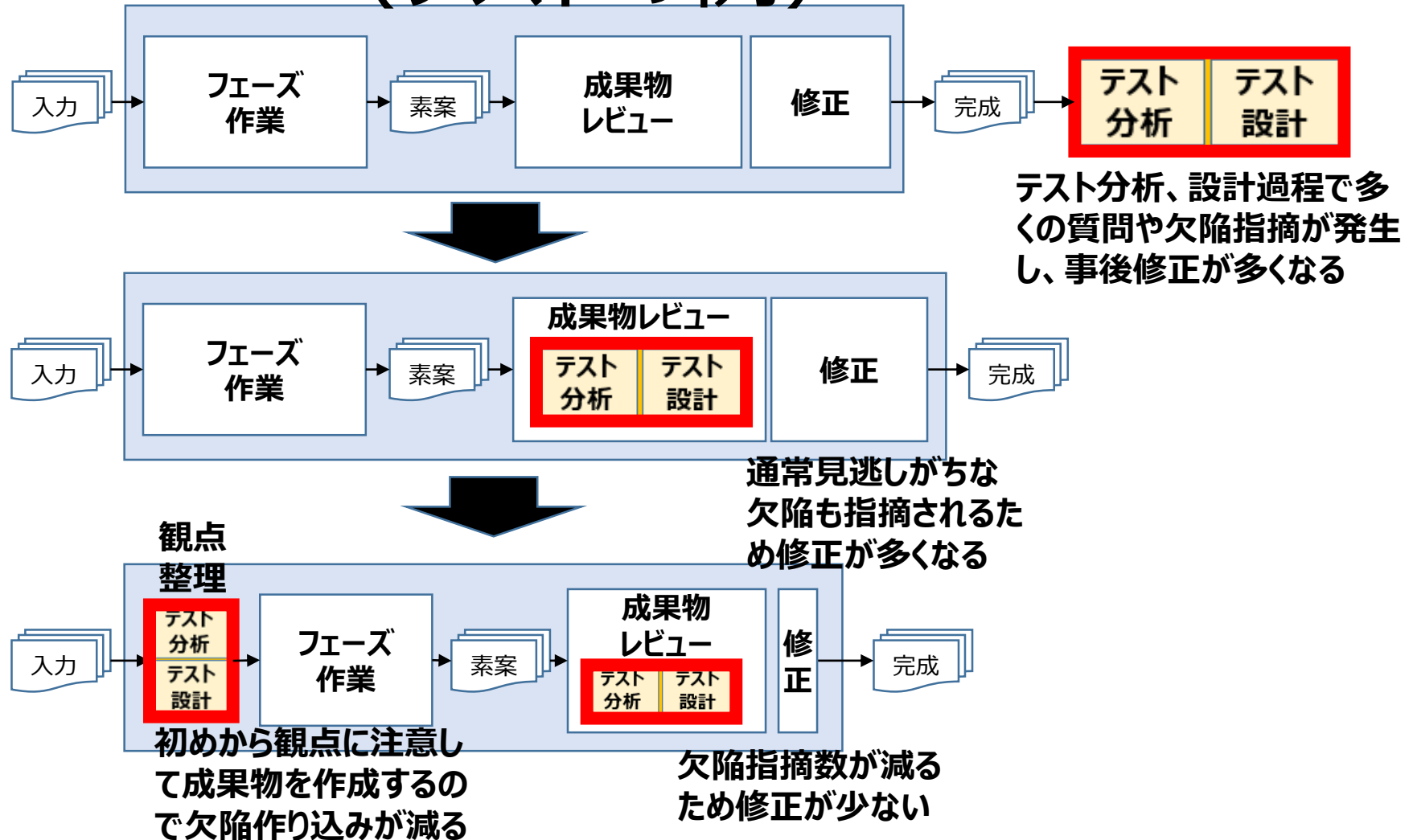
TDDのサイクル

1. 次の目標を考える
2. その目標を示すテストを書く
3. そのテストを実行して失敗させる (**Red**)
4. 目的のコードを書く
5. 2で書いたテストを成功させる (**Green**)
6. テストが通るままでリファクタリングを行う (**Refactor**)
7. 1~6を繰り返す



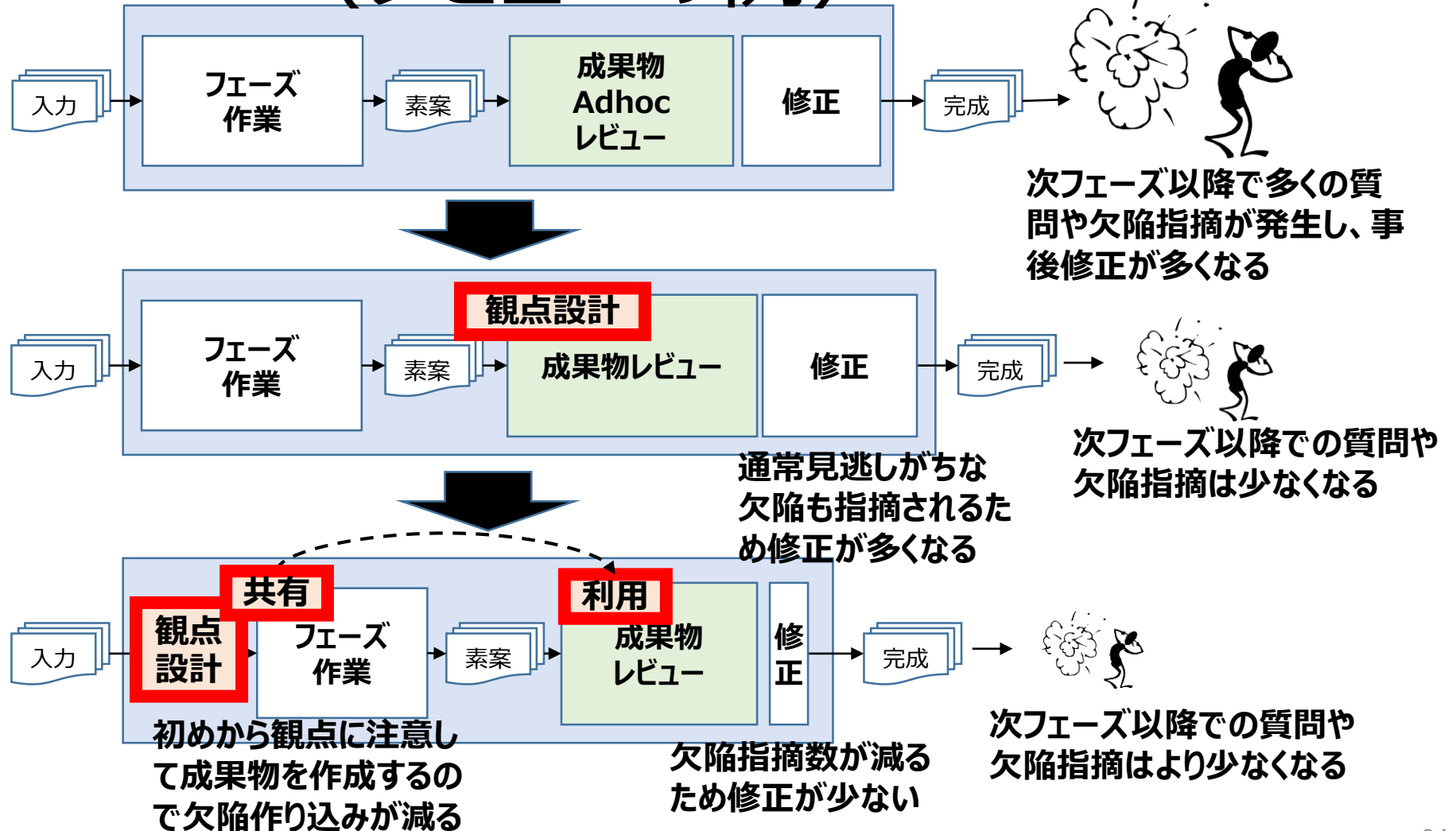
50分でわかるテスト駆動開発 /
TDD Live in 50 minutes より

同一フェーズ内でのシフトレフトライクなこと (テストの例)



**最初から品質
を作り込む**

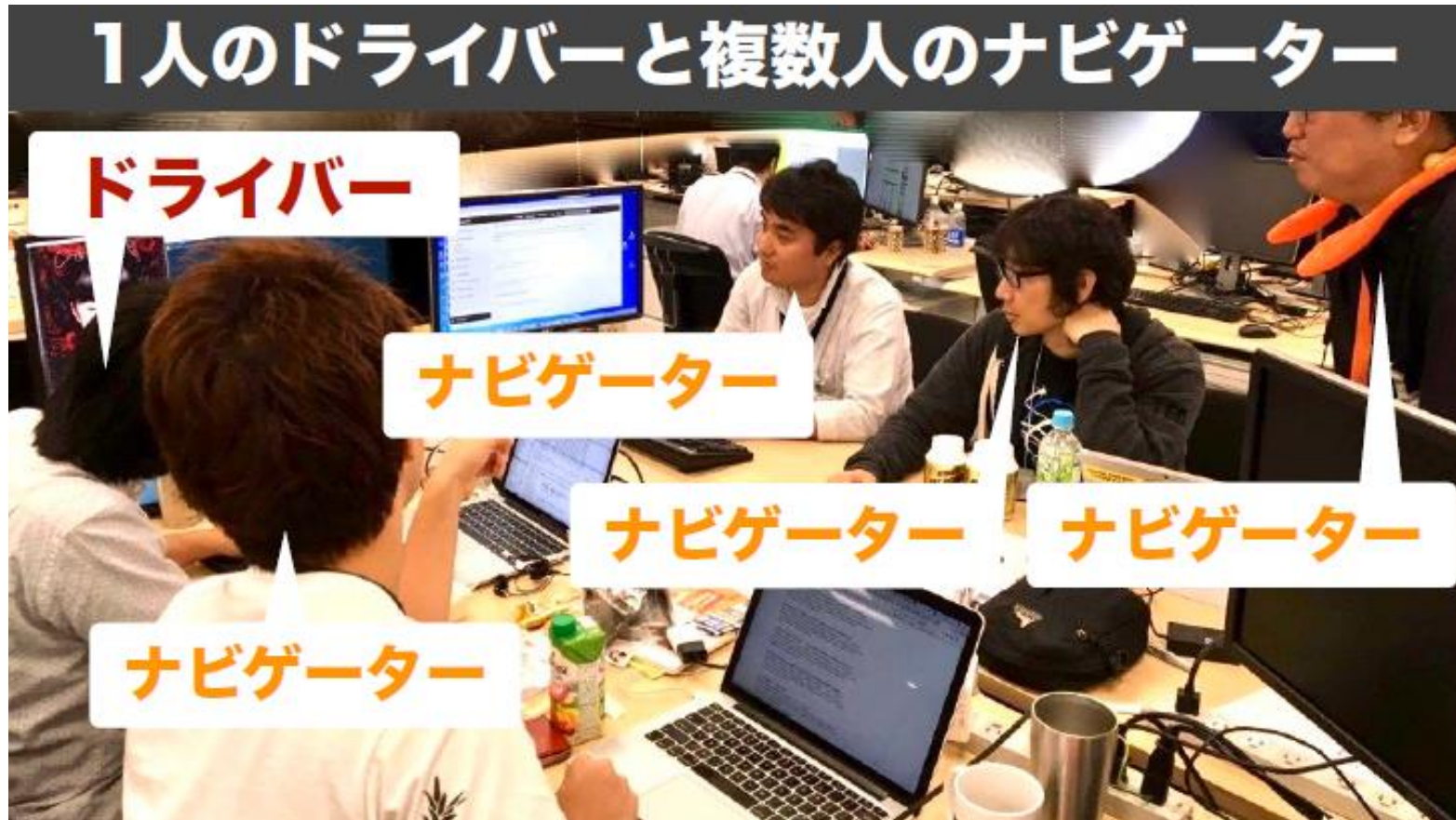
レビューにおけるシフトレフト **ライク**なこと (レビューの例)



**最初から品質
を作り込む**

モブプログラミング

JaSST Review2018 @TAKAKING22さん「レビュー再定義」講演資料より



ふりかえり

実施内容に対してふりかえり、結果を残しましょう

Keep:わかったこと・気づいたこと・感じたこと など	Try:今後の対応や方向性
Problem:現状の問題点や課題	

レビューアーキテクチャ

レビュー観点によるレビューアーキテクチャの設計

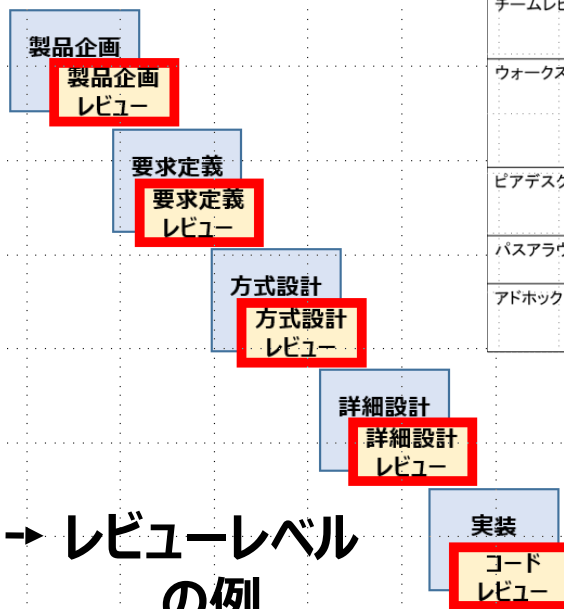
にしさんの予告 レビューアーキテクチャ

レビュー観点によるレビューアーキテクチャの設計

- どのようなタイプやレベルのレビューを行うことによって、どんな品質をどう保証するのか／どんなバグを検出するのか、の全体像を描く活動をレビューアーキテクチャと呼ぶ
 - マスターレビュープランやレビュー戦略と呼んでもよい
 - 個々のレビューのアクティビティ(≒レビュー会議)をレビューコンテナと呼ぶ
 - 例) ○○サブシステムに対する性能レビュー、○○サブシステムレビュー、性能レビュー
 - それぞれのレビューアクティビティでどのレビュー観点を分担するのか、を明示する
 - レビューコンテナ間の順序関係や依存関係を明示して、レビューの全体像を描き把握する

レビュー観点によるレビューアーキテクチャの設計

- どのようなタイプやレベルのレビューを行うことによって、どんな品質をどう保証するのか／どんなバグを検出するのか、の全体像を描く活動をレビューアーキテクチャと呼ぶ
 - レビューコンテナ内のレビュー観点、レビューコンテナで用いる技術や必要な技術レベル、レビューコンテナ内の(副次的)役割などをはっきりさせる
 - レビューレベル**: レビュー対象物の種類や粒度、順序関係などによる分類
 - 例) 仕様レビュー、アーキテクチャレビュー、詳細設計レビュー、コードレビュー、UMLレビュー、モジュールレビュー、派生部位レビュー、母体レビューなど
 - レビュータイプ**: レビューの意図や観点、技術などによる分類
 - 例) 静的レビュー、ウォークスルー、インスペクション、性能見積もり、セキュリティ設定確認、形式検証など
 - レビューサイクル**: 類似のレビューコンテナを繰り返す際の分類
 - 例) ○○レビュー第2回、やり直しレビューなど
 - レビューの(副次的)役割は組織ごとに色々多岐に渡る
 - レビューアーキテクチャとテストアーキテクチャを合わせたV&Vアーキテクチャを用いると、開発工程の進捗に合わせてどのよう品質が保証されていくのかが可視化できる
 - V&Vアーキテクチャが無いと、メトリクスやプロセスで「間接保証」せざるを得ず、大味になる



**レビューレベル
の例**
(テストレベルの対比)

**レビュータイプ
の例**
(テストタイプの対比)

レビュータイプ	概要	期待効果(欠陥除去)
インスペクション	最も厳格で、体系的、公式に実施されるレビュー。	1000行当たり16～17件という報告有り
チームレビュー	軽量化されたインスペクション。構造化されているがインスペクションほど公式、厳格でない。	1時間当たりの欠陥検出数はインスペクションの2/3という報告有り
ウォークスルー	作成者が主導して実施されるレビュー。作成者自らが対象成果物内容を説明することで	1000行当たり8件程度(インスペクションの半分)という報告有り
ピアデスクチェック		欠陥検出する
パスアラウンド	作成者が作業成果物のコピーを複数人に配付し、複数のコメント・フィードバックを獲得する方法。	(比較的時間がある場合や分散開発で有効)
アドホックレビュー	場当たり的、思いつきで実施するレビュー。ある特定の切り口に対する意見・コメントを即興で求める場合にのみ有効。	欠陥除去にはあまり寄与しない

- ☐ 誤字・脱字・衍字チェック
- ☐ 曖昧・不明確・矛盾チェック
- ☐ 規約・様式遵守チェック
- ☐ 利用時シナリオ分析
- ☐ インターフェース分析
- ☐ 合目的性確認
- ☐ リスクマネジメントレビュー
- ☐ 性能見積レビュー
- ☐ セキュリティ要件実装レビュー
- ☐ 内部・外部整合性確認 など

SQIPシンポジウム2019 併設チュートリアル3
「ソフトウェア開発における品質の作り込み ～フロントローディングの基礎～」より

レビューアーキテクチャ検討例

「要求仕様書のレビュー技法と実践のポイント」安達2020 より

ドキュメント特性

基本・形式

曖昧

規約
不遵守

不明確

誤字
脱字
衍字

矛盾

未決

不統一

判読性

NG表現

多義表現

抽象表現

レビュー共通(レビュー自動化へ)

どのドキュメントでも必ず最初
にチェックする共通確認事項

サービス領域(利用)

有効性

効率性

利用状況
適合性

リスク
回避性

サービス領域(提供)

運用性

信頼性

保守性

システム／ソフトウェア領域

機能

合目的性

正確性

適切性

セキュリティ

非機能

性能

信頼性

保守性

使用性

互換性

移植性

画面

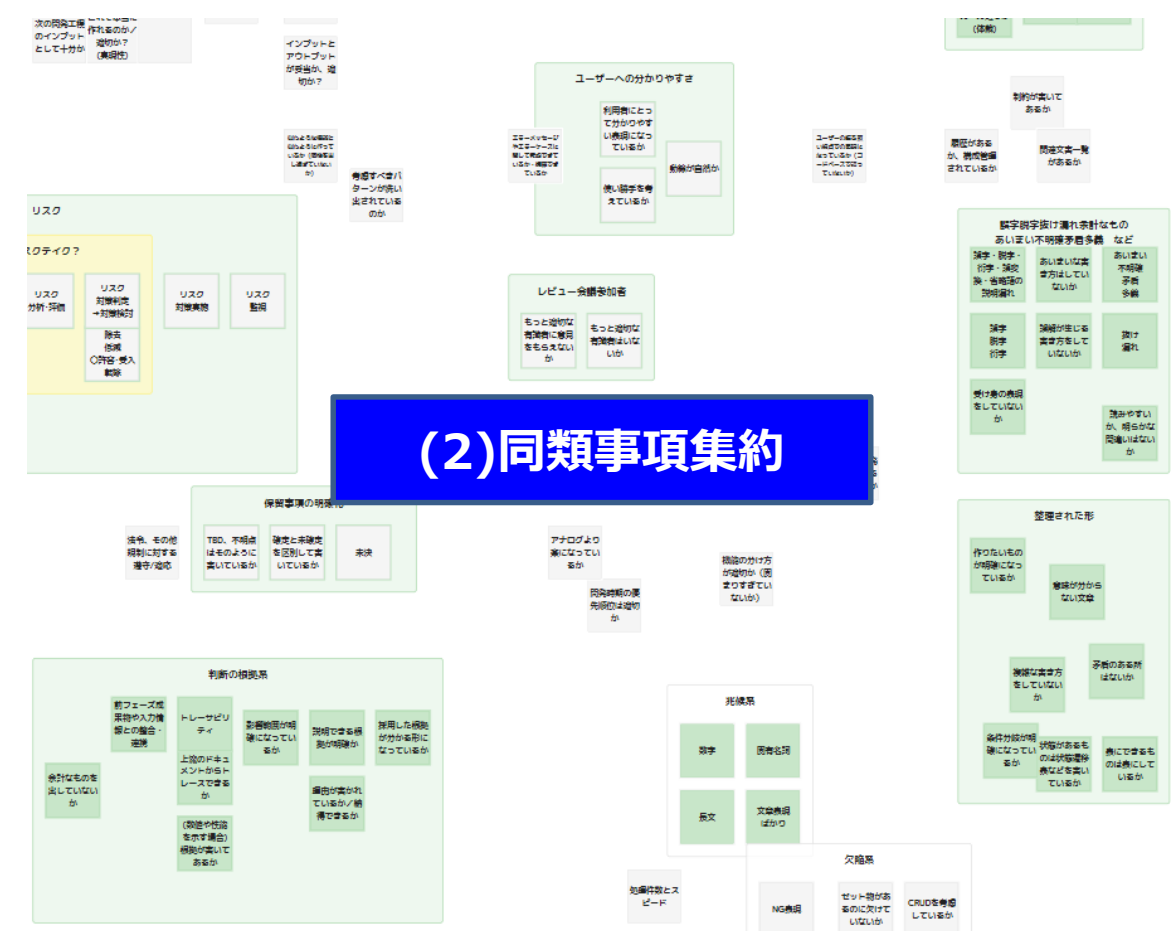
構造/論理

製品・サービスの特徴とフェーズにより取捨選択

開発フェーズ進行に応じて
選択してチェックする確認事項

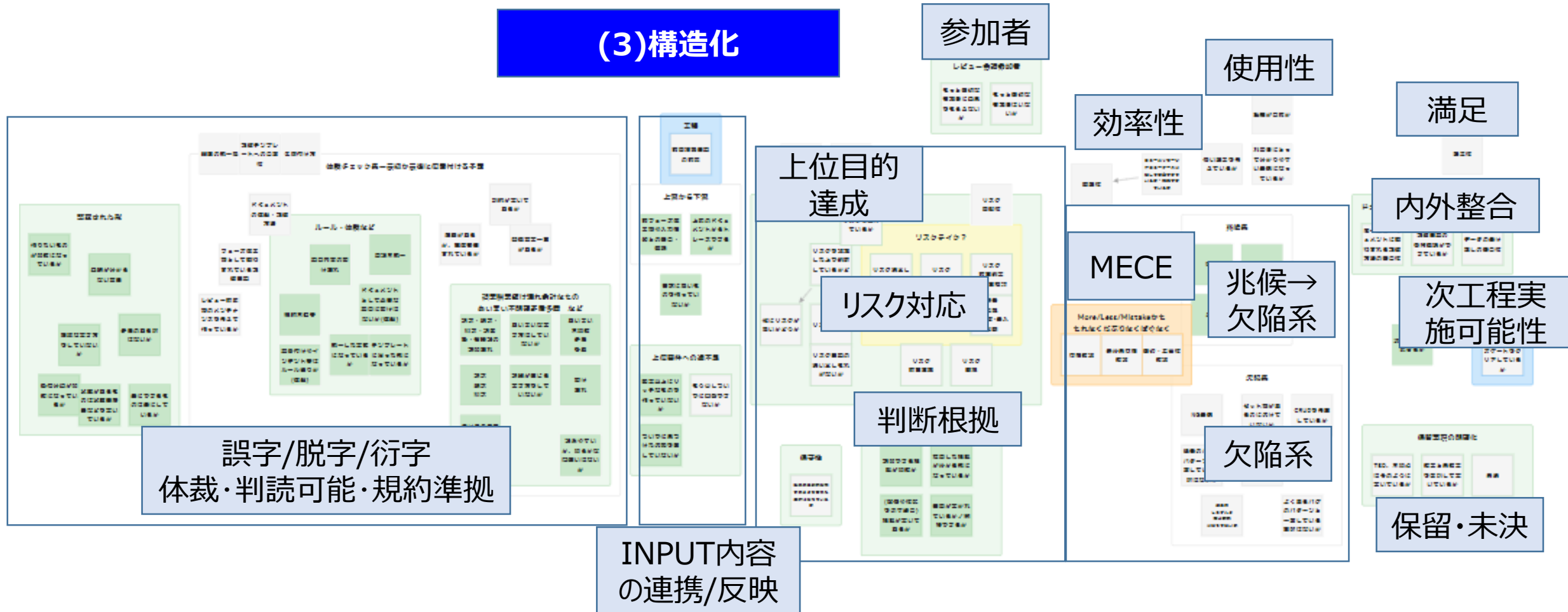
実行委員によるレビューアーキ構築試行結果

(1)観点洗い出し $\longleftrightarrow$ (2)同類事項集約 $\longleftrightarrow$ (3)構造化

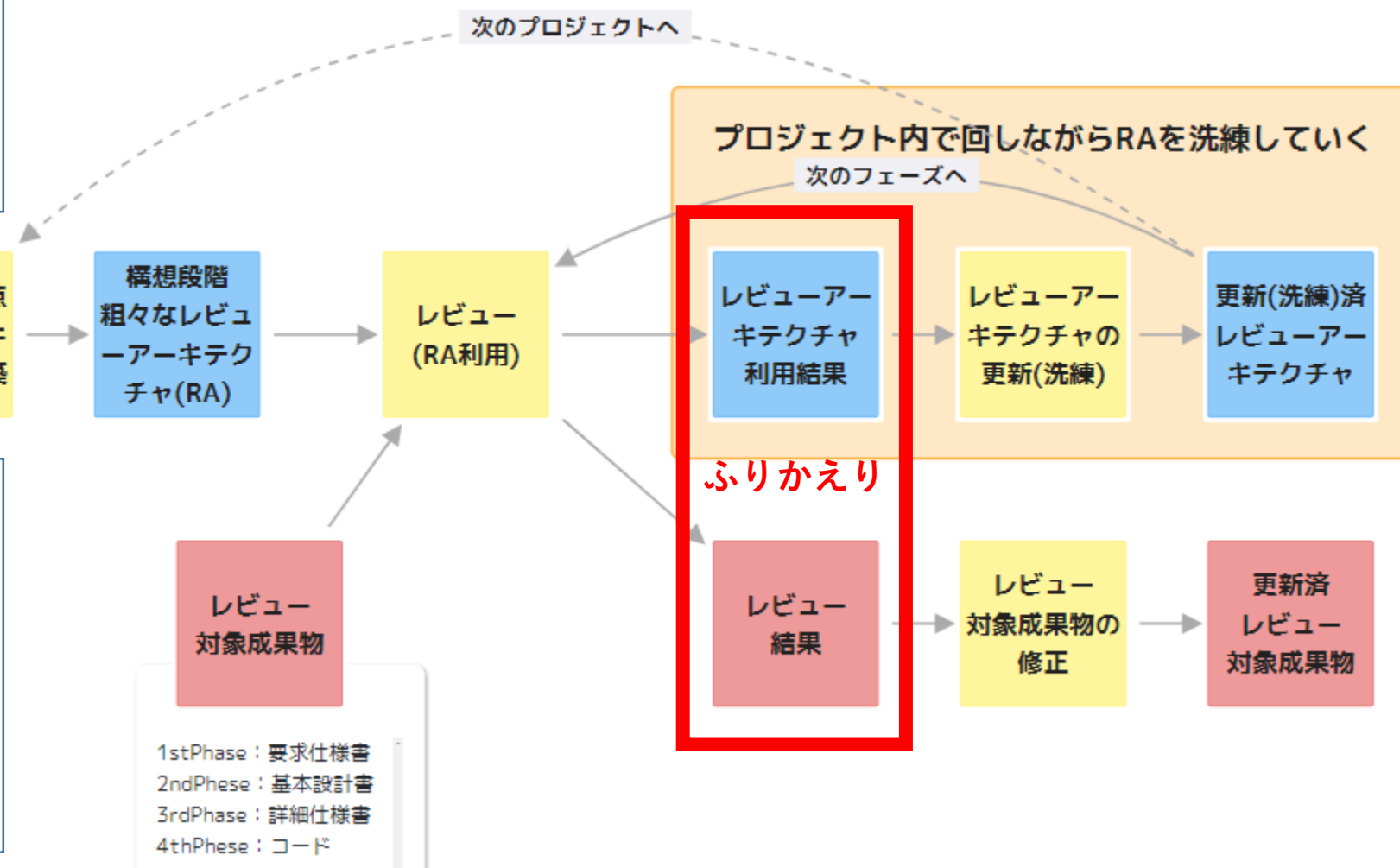
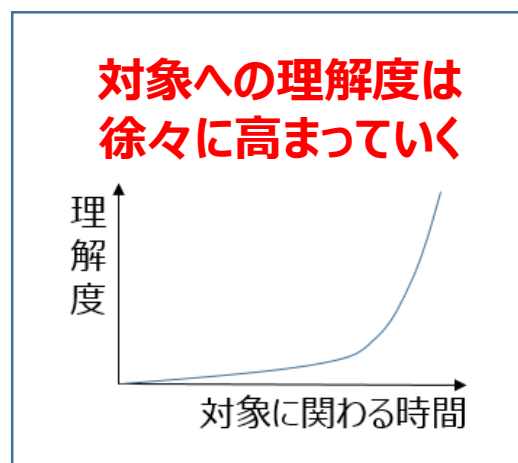
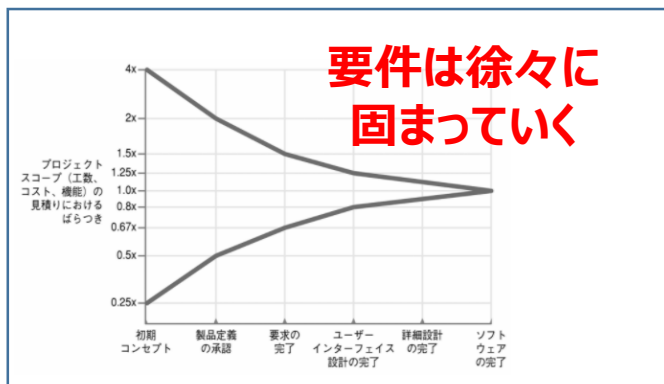


実行委員によるレビューアーキ構築試行結果

(1)観点洗い出し $\longleftrightarrow$ (2)同類事項集約 $\longleftrightarrow$ (3)構造化

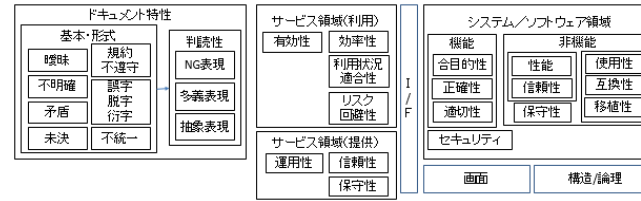


レビューアーキテクチャはプロダクト開発サイクルを回しながら段階的に洗練していく



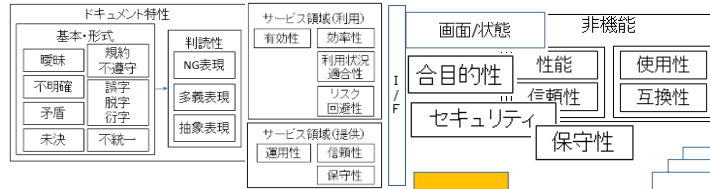
レビューアーキテクチャの利用方法（例）

Aプロダクト用レビューアーキテクチャ



Tailoring（取捨選択）

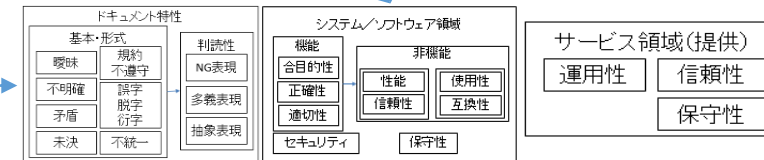
要求定義Phase用レビュー観点群



過去トラブル実績



システムデザインPhase用レビュー観点群



要求定義

要求仕様書
Review

要求仕様書

Review
結果

システム
デザイン

設計書
Review

ふりかえり

実施内容に対してふりかえり、結果を残しましょう

Keep:わかったこと・気づいたこと・感じたこと など	Try:今後の対応や方向性
Problem:現状の問題点や課題	

まとめ

レビュー観点を活用して モノづくりとその確認の両面で成長していきましょう！

No.	レビュー観点の効果と必要性
1	"観点"があれば雑多な情報群の中から <u>集中して該当を直接探し当てられやすい</u>
2	"観点"がない場合、 <u>書かれていることだけに反応した指摘に終始してしまう</u>
3	有効な指摘事項など <u>優先度の高い欠陥・不備を見つけやすい</u>
4	"観点"がない場合、メンバー間での <u>重複確認</u> や <u>誰も見ていない抜け・漏れ</u> が発生しやすい
5	"観点"があれば、 <u>メンバー間で確認する観点の分担が可能になる</u>
6	<u>有効指摘のノウハウを再利用しやすくなる</u> （ <u>有識者依存度を低減できる</u> ）

お疲れさまでした

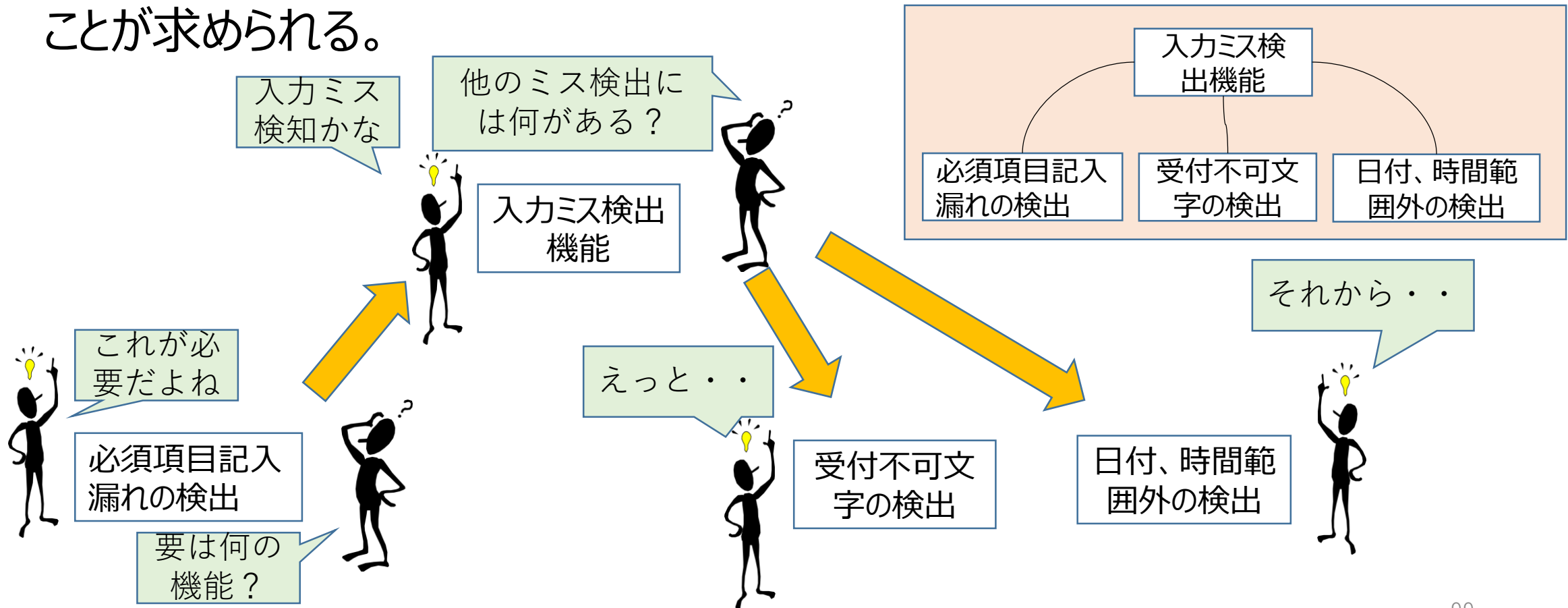
【付録】

当アプローチの実践上の注意事項

目的～観点構造化は重要だが簡単ではない

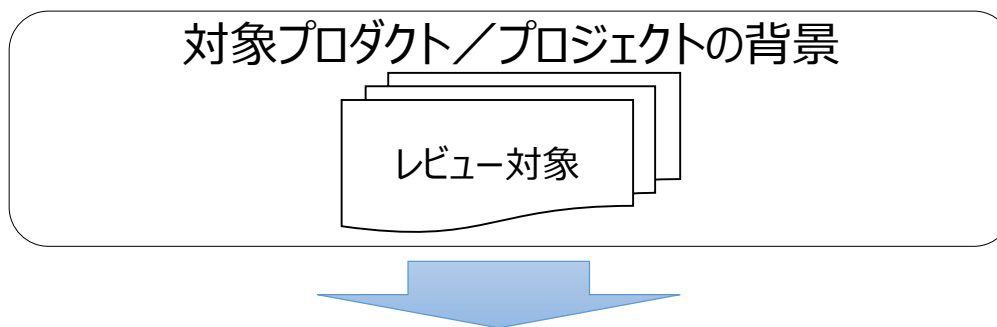
構造化・モデル化にはノウハウがある

- 観点設定ではトップダウン／ボトムアップの両面アプローチとMECEであることが求められる。



同じ対象でもチームによって観点設定がばらつく

観点設定の適切性レビュー&フィードバックが必要



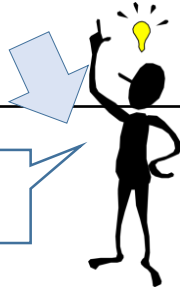
<各グループ検討の結果（例）>

グループ	Gr1	Gr2	Gr3	Gr4	Gr5	Gr6
設定した レビュー目的	最低限実現可能 かどうか確認する	技術的实现性の 課題把握	要求の適切性見 極め	機能把握・機能 必要性検討	空欄	顧客に価値ある 商品かを確認
設定した レビュー観点	成立性 実現性 利便性	・明確さ ・デフォルト妥当 性 ・成立性 ・操作実現性	・安全性 ・使用性 ・機能性 ・実用性	・機能実現性 ・使いやすさ ・安全性	・異常時の動作 ・あいまい表現 ・ユーザ視点 ・費用対効果	・エンドユーザ視 点 ・開発者視点 ・テスト視点 ・マーケティング視 点

人によって適切に実践できる観点粒度が異なる

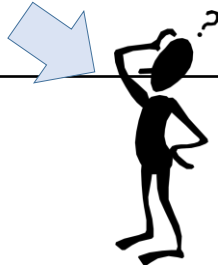
レビューの状態から観点粒度を調整する

利害関係者	対象システムにおける主な活動	関心事	必要なレビュー観点・対象	レビュー観点の具体化
実利用者	システムに必要情報を入力して、ほしい結果を得る	誤入力情報がそのまま処理されないか？	入力ミス検知機能があるか？（入力画面）	☐ 必須項目記入漏れの検出 ☐ 入力不可文字の検出 ☐ 日付、時間範囲外の検出

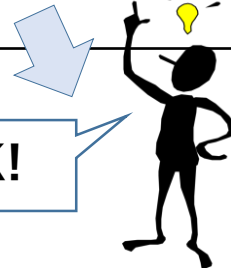


OK!

Aさん



Bさん

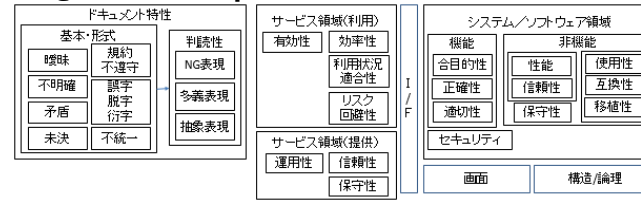


OK!

Bさん

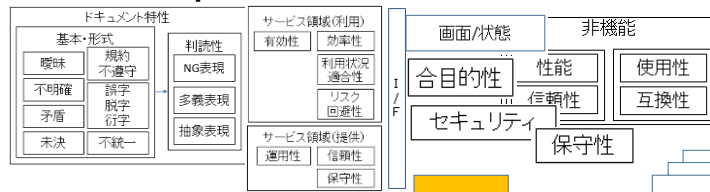
いつもゼロから実践すると費用対効果が低い 実践経験と実績の蓄積→分析結果の活用により打開できる

QA Viewpoint for Product



Tailoring

Viewpoint for RD Phase



Troubles from the past

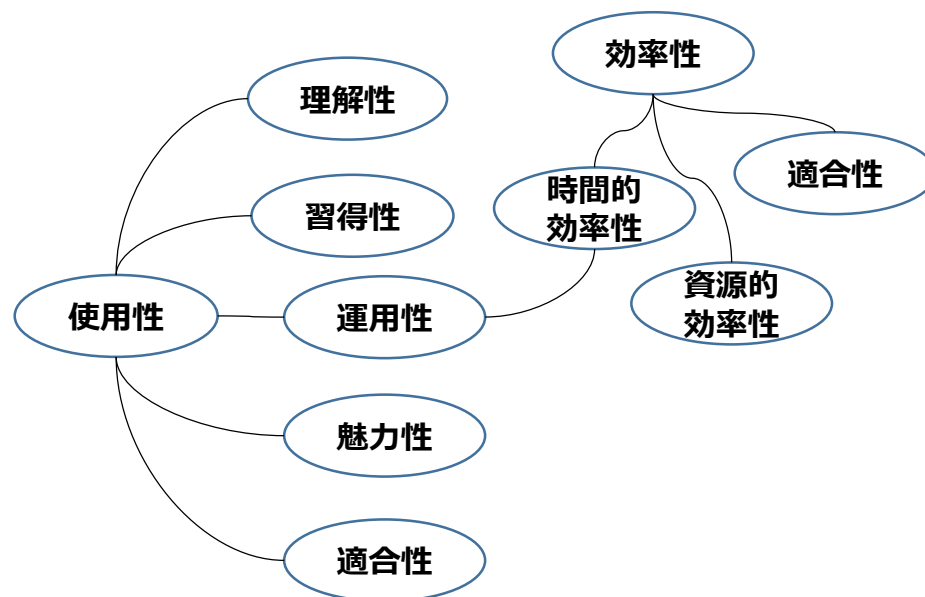


レビュー観点のまとめ方は組織の腕の見せ所

どこまでを対象範囲とするか：特定プロダクト～組織共通

どのような表現にするか：文章・一覧表・マトリクス・モデル図など

	機能性	信頼性	使用性
要求定義	△	△	○
基本設計	○	○	
詳細設計	○		
実装	○		○



確認観点リスト		
	機能要求確認	☒
	非機能要求確認	☒
	別モデル化確認	☒
	外部内部整合性	☒
		☐
		☐
		☐
		☐

共有するレビュー観点表(図)は継続的なメンテナンスが必要。

多くの組織がこの実践を苦手としている。

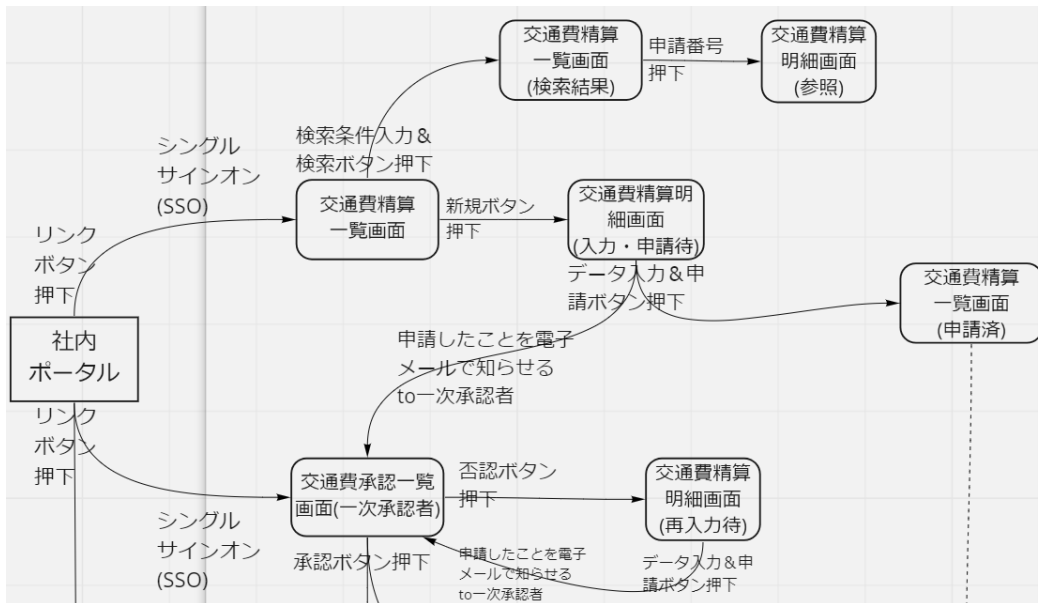
絶対にうまくいく唯一の方法はない～ふりかえりによる継続的改善が必要

必要なレビュー観点導出技法の取り揃え

レビュー研究・事例情報等を参考に自ら開発しラインナップ化

レビュー観点導出技法（A7） 候補

対象成果物の記述内容を、別モデル（例1：フローチャート、例2：状態遷移図・状態遷移表、例3：マトリクス表）で表現し直すことでレビューを実施する。※どのモデル表現を採用するかを決める＝レビュー観点導出とする



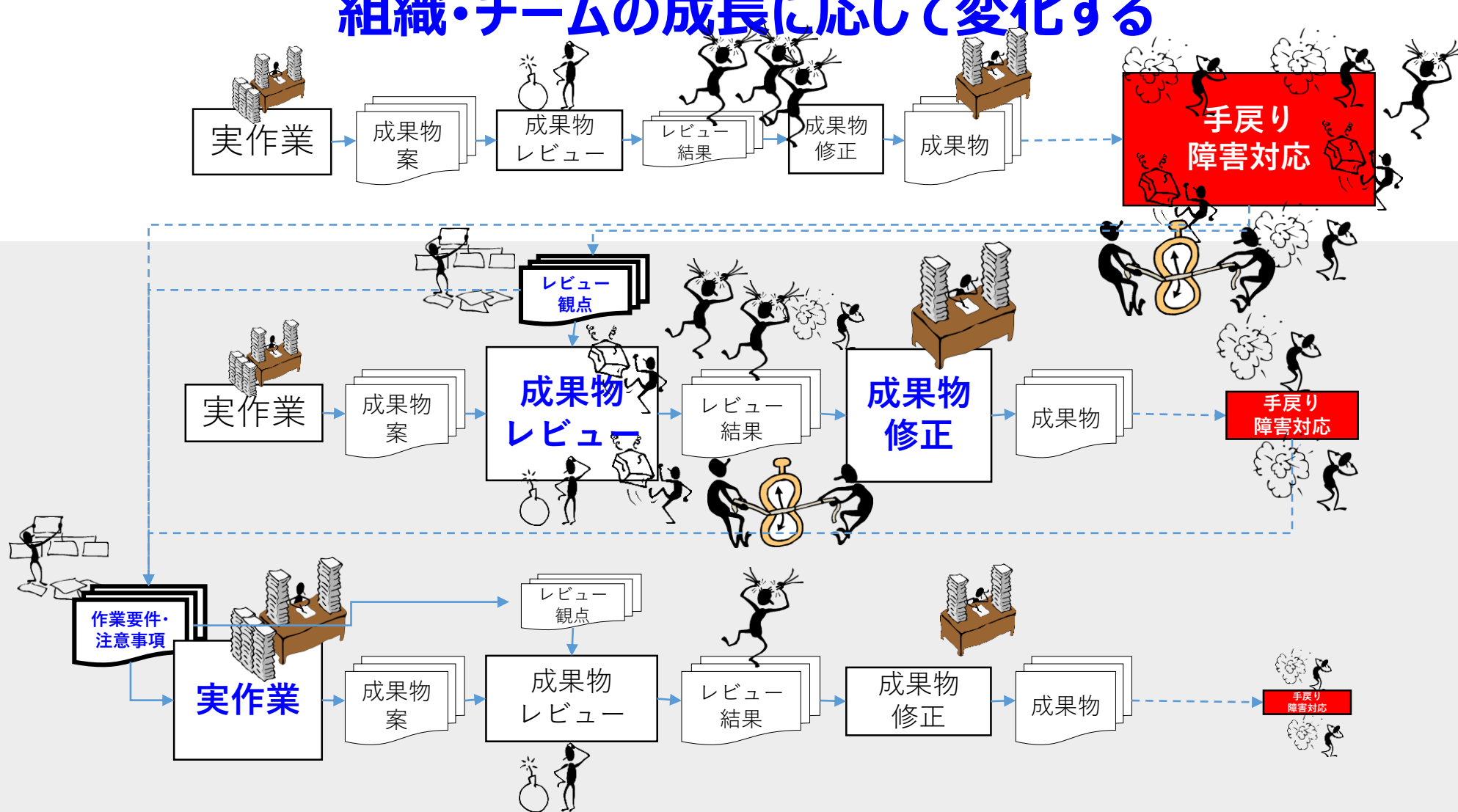
事例1：状態遷移図

入力・遷移元・イベント	判定・処理	出力・遷移先
事前条件		事後条件

事例2：テストケースフレーム

レビュー観点導出のタイミング

組織・チームの成長に応じて変化する



参考文献

- Software Quality In 2008(JaSST'08東京) Capers Jones
<https://www.jasst.jp/archives/jasst08e/pdf/A1.pdf>
- レビュー目的・観点設定の効果と課題(JaSST'16東京)
<http://www.jasst.jp/symposium/jasst16tokyo/pdf/A2-1.pdf>
- ソフトウェアレビュー研究結果の認知拡大と適用促進(SS2022)
<https://www.sea.jp/ss2023/download/16-ss2023.pdf>
- 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(JaSST'19北陸) 補足資料
<http://www.jasst.jp/symposium/jasst19hokuriku/pdf/S2-2.pdf>
- ソフトウェアテストシンポジウム2022東京 ワークショップ
そゆことね！よくわかるレビューテクニック～明日から使える技術をSQiPレビュー研究会からあなたに～
<https://www.jasst.jp/symposium/jasst22tokyo/pdf/F6.pdf>
- “レビュー体系化”の経過報告 レビュー体系とレビューアーキテクチャ(JaSST Review'23)
<https://www.jasst.jp/symposium/jasstreview23/pdf/S1.pdf>
- 引用：要求工学：第3回要求仕様
<https://www.bcm.co.jp/site/2004/2004Dec/04-youkyuu-kougaku-12/04-youkyuu-kougaku-12.htm>
- 観点活用レビューワークでわかったこと 一意な観点設定から観点設計への壁(SQiP2024)