

レビュー観点の抜け漏れ防止に向けた  
レビュー要求分析・設計・  
実装試行でわかったこと

株式会社 HBA 安達賢二

<http://www.software-quasol.com/>  
adachi@hba.co.jp

# 背景

得できない

いる

ユーの  
プットが  
不明確

インスペクション  
をやるほど工数  
を持っていない  
場合も多い

実施調整ができ  
ずサーバに成  
果物案を置き見  
ておいてね、に  
なる

誤字脱字が  
多い成果物  
案も多い

レビューす  
るに値しな  
い成果物も  
多い

レビュー工数  
割り当てが  
少ない

レビュー  
する暇が  
ない

レスポンスが  
ないと問題な  
しと受取るこ  
ともある

仕様や設計  
の記載の仕  
方の問題も  
ある

人によりコ  
ド等の書  
方が異な

なんでもレ  
ビューすること

**多くの方たちが  
レビューにはいろいろ問  
題がある、効果が実感で  
きない、と言っています。**

どの場面での  
レビュー形式を  
採用したらよい  
のかが不明確

議論したが仕様  
に未記載で設計  
に考慮されない

レビュー指標：  
上限値・下限値  
はあるが、当て  
はまらないこと  
も多い

レビュー生  
産性が低い

バグ分析で必  
要な作業が抜  
けていること  
が分かる

レビューの  
効果が不明瞭

チェックリス  
トが数100  
件で大変

ハーのレベ  
ル不ぞろい

明確な誤り以外  
指摘しない、欠如、  
過剰は見逃され  
る傾向あり

“ざるレ  
ビュー”に  
なっている  
ことがある

漏れなど同  
の指摘が複  
数のプロジェクト  
で発生する

時間がか  
かる・効率  
が悪い

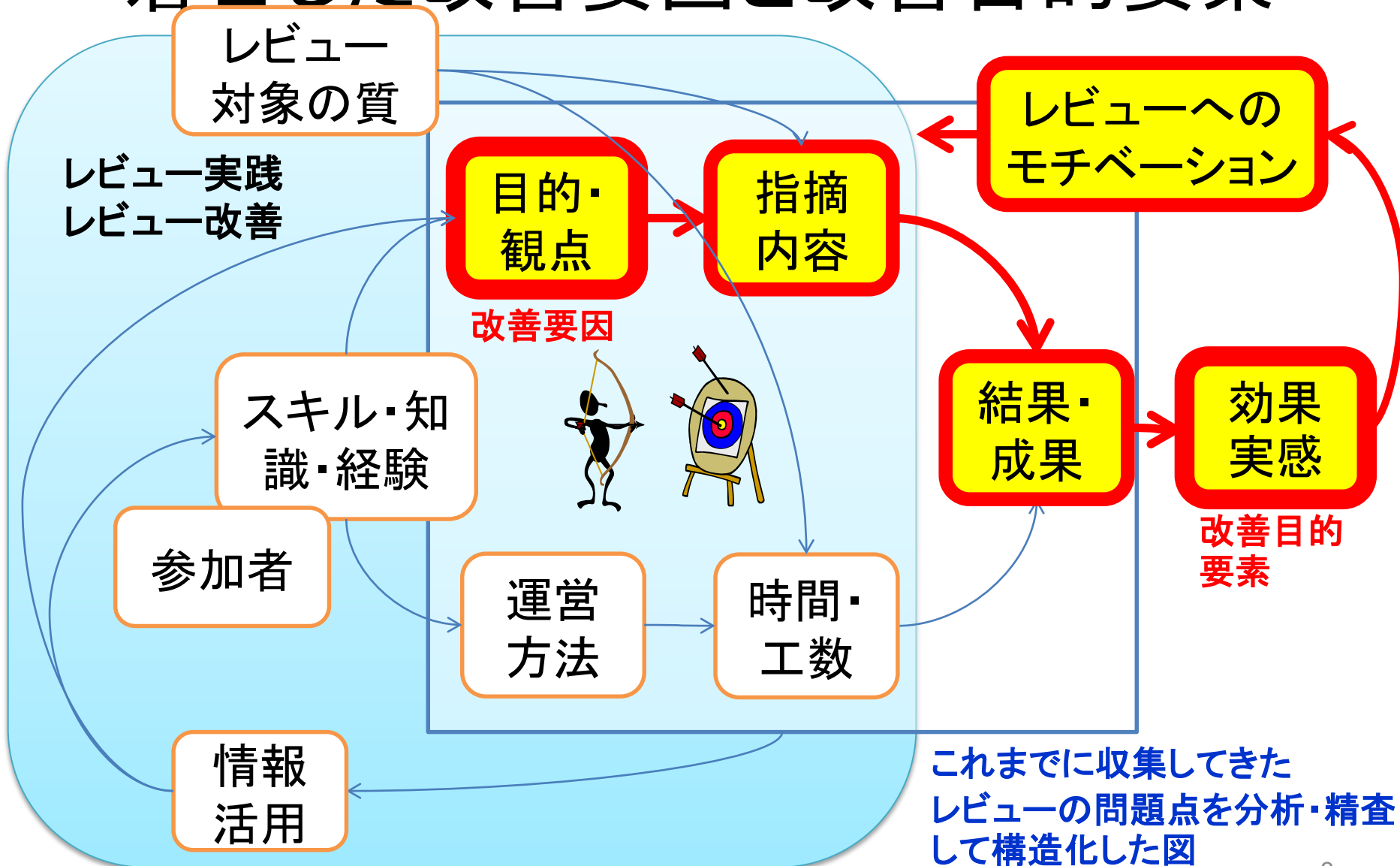
対象ドメイ  
ンの有識  
者不足

すぐ目に見え  
るところだけレ  
ビューする傾  
向がある

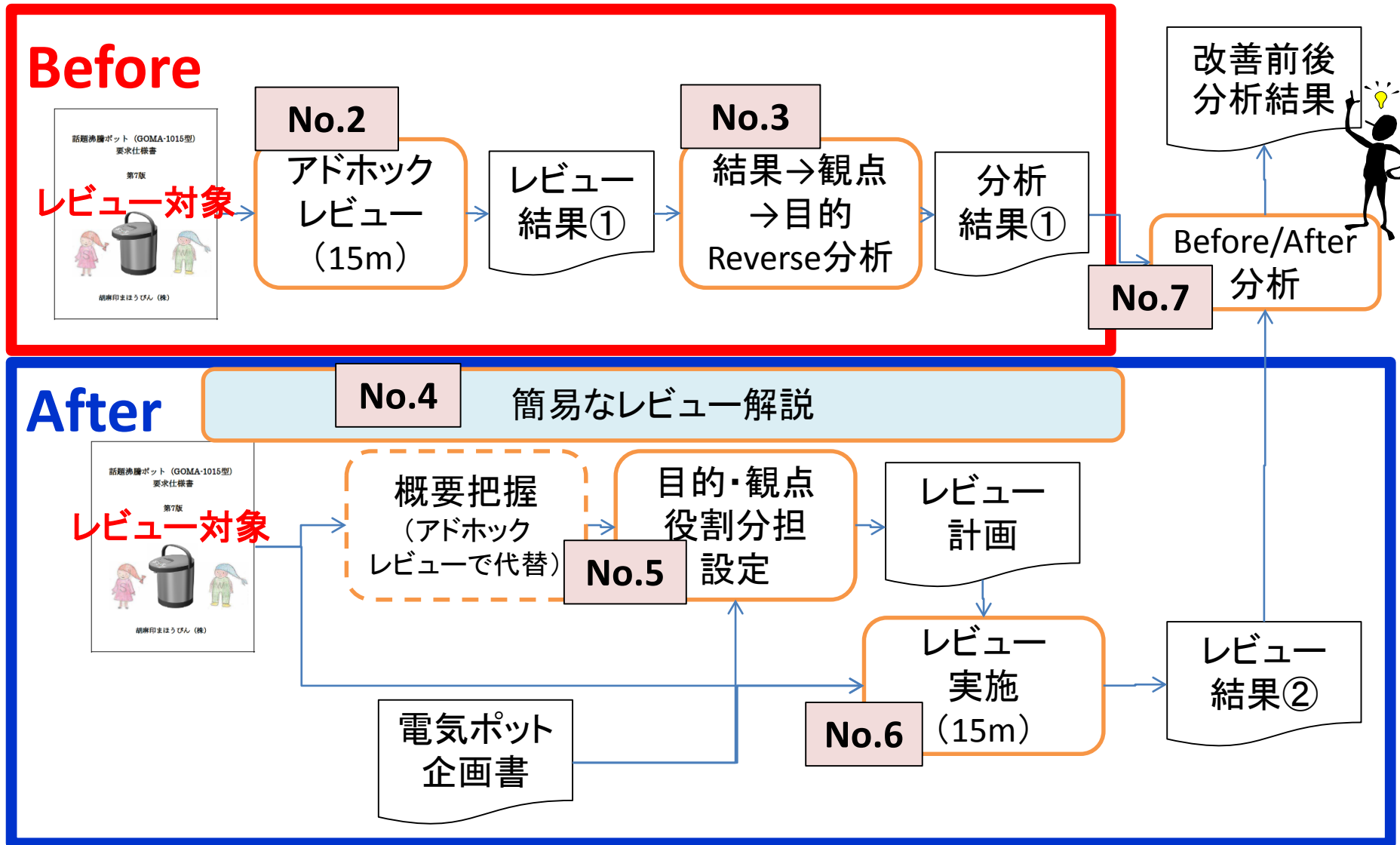
人を沢山投入  
しても欠陥指  
摘が少ない

出典: 森崎、野中、安達: ソフトウェア品質シンポジウム2009 企画セッション  
『レビューの壁を破る』 +SQiP2013-SIG 参加者が持つレビューへの問題意識

# レビューの問題構造のうち 着目した改善要因と改善目的要素

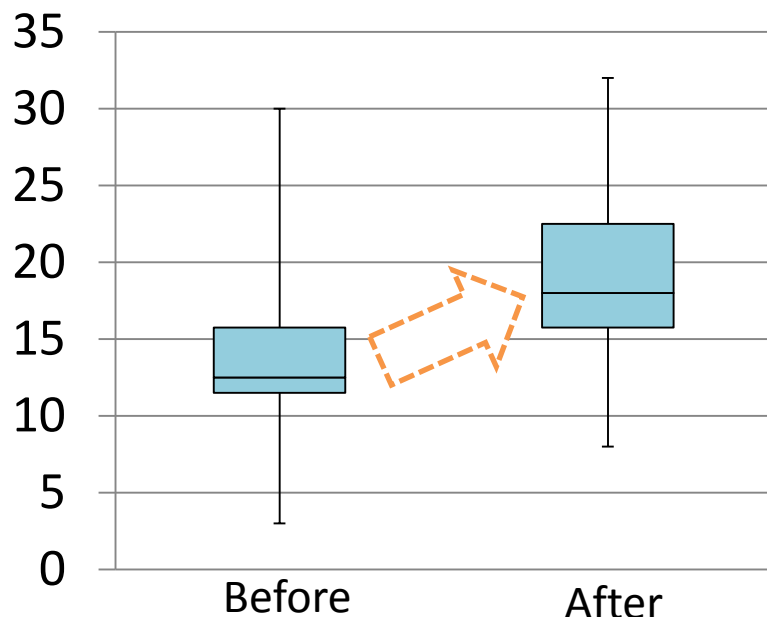


# 改善目的達成を促進するソリューション レビュー実践ワークショップ概要



# 2012年～2015年に実施・分析可能なデータが存在した ワークショップ実施済10社24チームの結果

## 指摘件数の変化

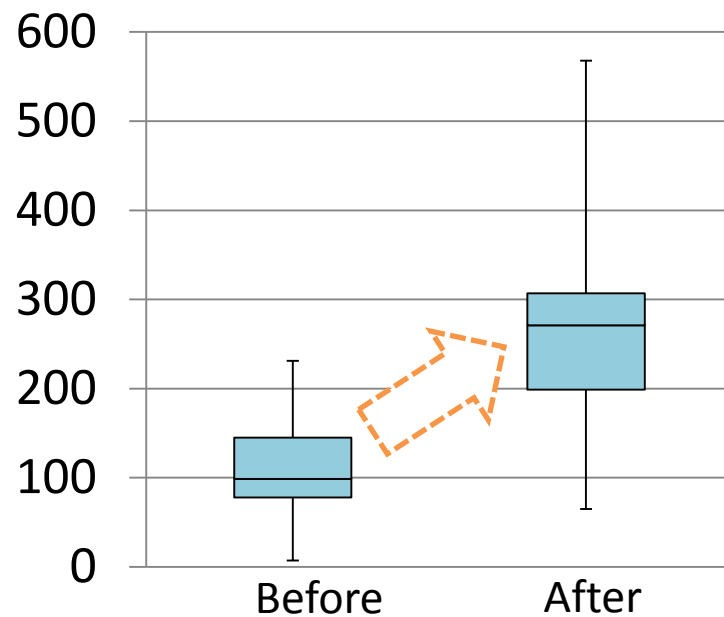


- ・指摘件数増: 18チーム (Max20件増)
- ・指摘件数変化なし: 2チーム
- ・指摘件数減: 4チーム (Max13件減)
- ・指摘件数変化平均: 4.8件増／チーム

平均4.8件増

中央値5.5件増

## 指摘内容(価値)の変化



- ・指摘価値増: 21チーム (Max424p増)
- ・指摘価値減: 3チーム (Max58p減)
- ・指摘価値変化平均: 151.5p増／チーム

平均151.5P増

中央値172.5P増

# 指摘価値変化例

指摘件数:  $\Delta 23 \rightarrow \bullet 24$

<p><math>\Delta</math>改善前 (Before)</p> <p><math>\bullet</math>改善後 (After)</p>		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	<p>効果大</p> <p>主対象: 要件抜け・誤り</p> <p>5</p>					<p>30 ↓ 335</p>
	<p>効果中</p> <p>主対象: 機能上のバグ (誤植による)</p> <p>3</p>	<p><math>\Delta</math>が左下 (指摘価値が低い)に集中</p>		<p><math>\bullet</math>が右上 (指摘価値が高い)に集中</p>		<p>75 ↓ 93</p>
	<p>効果小</p> <p>主対象: 誤字・脱字・衍字 規約違反</p> <p>1</p>					<p>16 ↓ 8</p>
計		35→19	18→0	40→340	28→77	121→436

# 次の課題

解決に向けた第一歩は踏み出したものの、次にクリアすべき課題があります。

# 最上位観点のバラつき

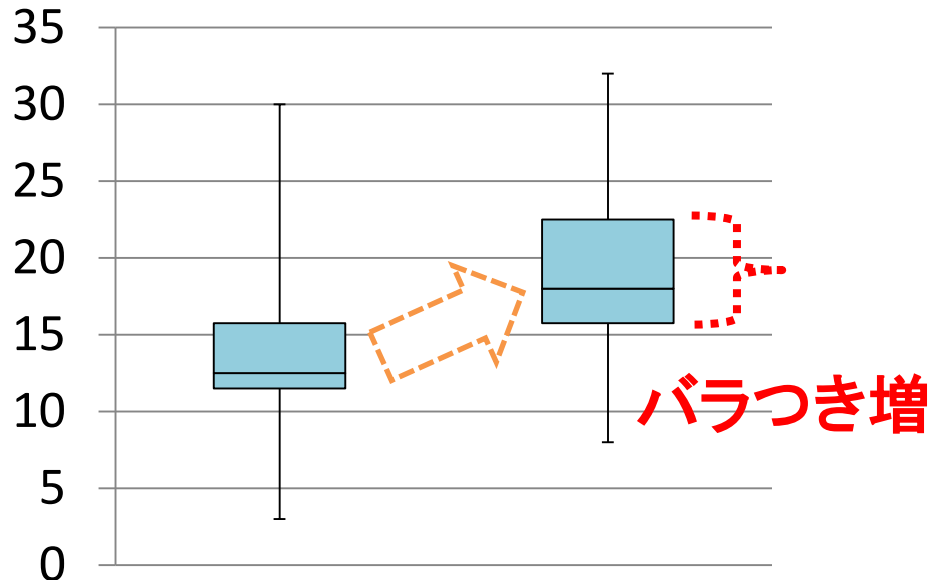
チーム1	チーム2	チーム3
<input type="checkbox"/> コスト <input type="checkbox"/> 操作性 <input type="checkbox"/> 安全性 <input type="checkbox"/> 性能	<input type="checkbox"/> 簡単なのか <input type="checkbox"/> 経済的なのか <input type="checkbox"/> 安全なのか <input type="checkbox"/> 訴求力はあるか <input type="checkbox"/> 開発が容易か	<input type="checkbox"/> 企画満足 <input type="checkbox"/> 操作性 <input type="checkbox"/> 安全性 <input type="checkbox"/> 性能(数値目標)
チーム4	チーム5	チーム6
<input type="checkbox"/> 企画満足 <input type="checkbox"/> ユーザに適切な機能か <input type="checkbox"/> 機能実現方法が明確か <input type="checkbox"/> ドキュメント矛盾はないか	<input type="checkbox"/> 企画満足 <input type="checkbox"/> システムテストができる内容か <input type="checkbox"/> 設計ができる内容か？ <input type="checkbox"/> ドキュメントの一貫性、整合性 <input type="checkbox"/> 安全性	<input type="checkbox"/> 使いやすさ <input type="checkbox"/> 経済的か <input type="checkbox"/> 安全性

観点設定がなければ、レビューで確認されない



# 個別確認項目のバラつきが大きいことも想定される

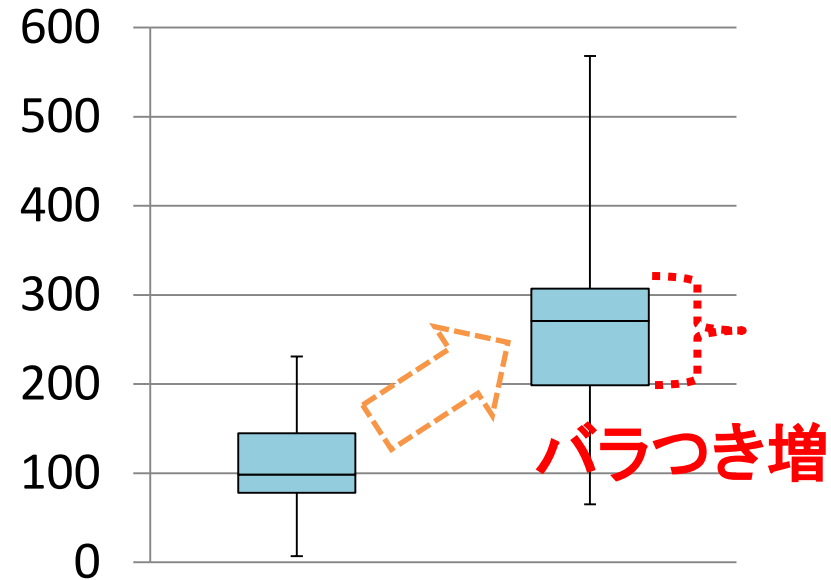
## 指摘件数の変化



- ・指摘件数増: 18チーム (Max20件増)
- ・指摘件数変化なし: 2チーム
- ・指摘件数減: 4チーム (Max13件減)
- ・指摘件数変化平均: 4.8件増/チーム

平均4.8件増  
中央値5.5件増

## 指摘内容(価値)の変化



- ・指摘価値増: 21チーム (Max424p増)
- ・指摘価値減: 3チーム (Max58p減)
- ・指摘価値変化平均: 151.5p増/チーム

平均151.5P増  
中央値172.5P増

# 課題への対策

課題への対策として以下のような概念を導入するとどうなるかやってみました。

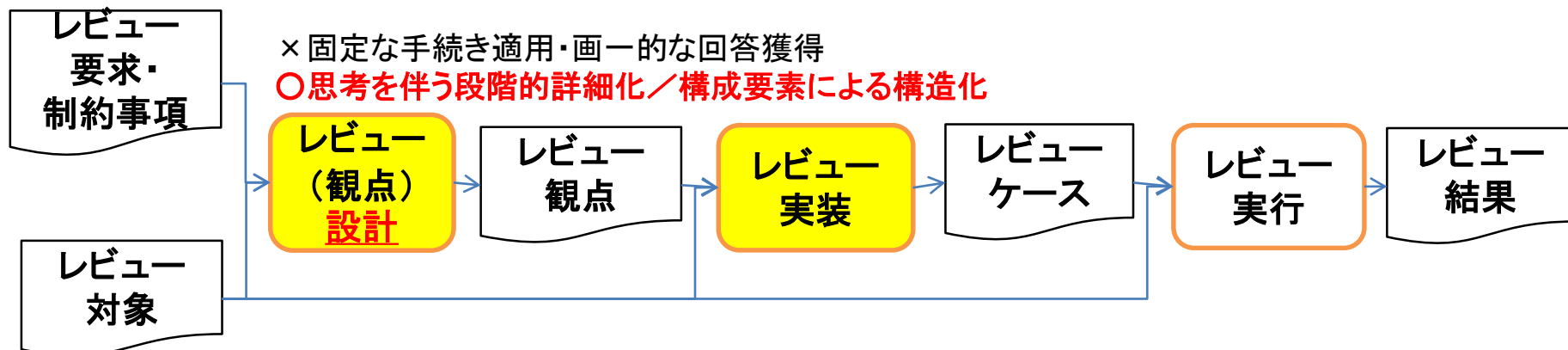
## レビュー開発・保守プロセス

レビュー  
要求分析

レビュー  
設計

レビュー  
実装

レビュー  
実行



# 立ち位置からの観点導出

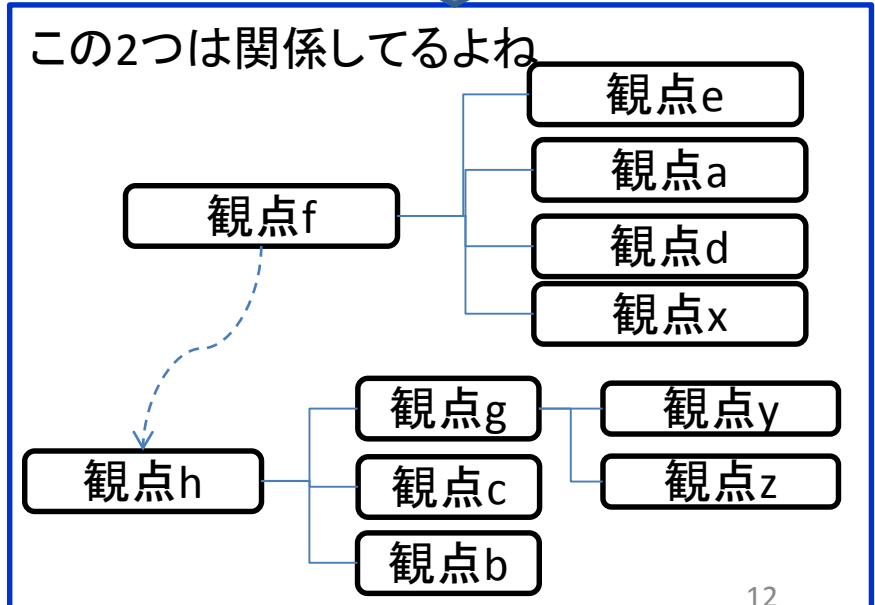
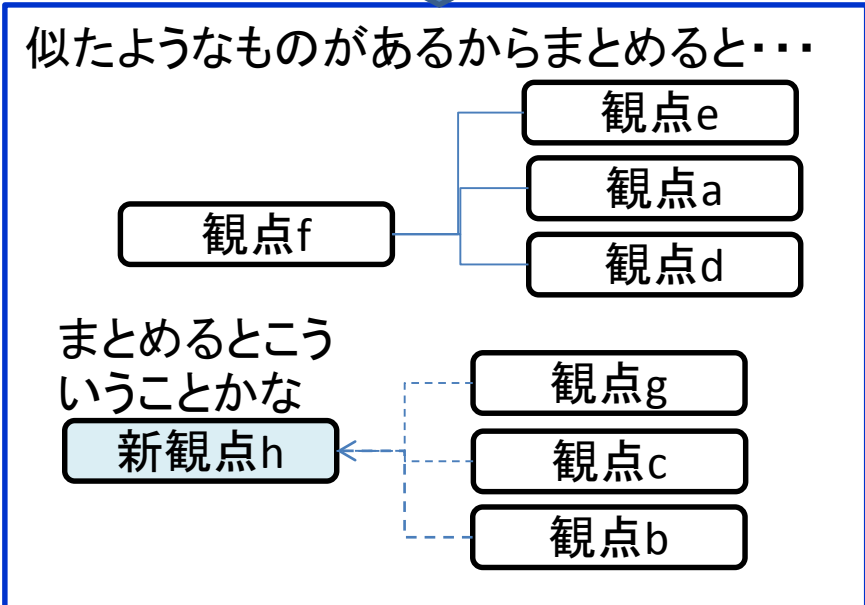
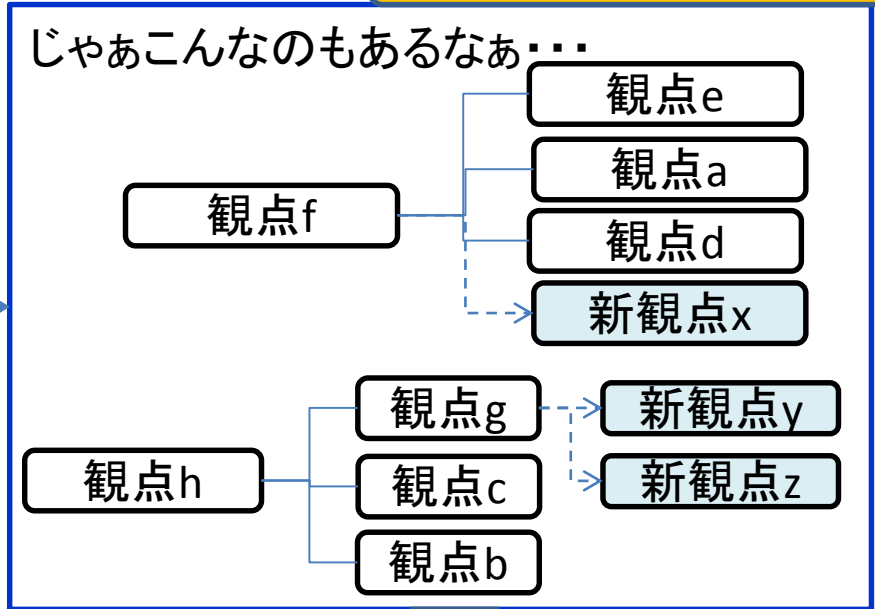
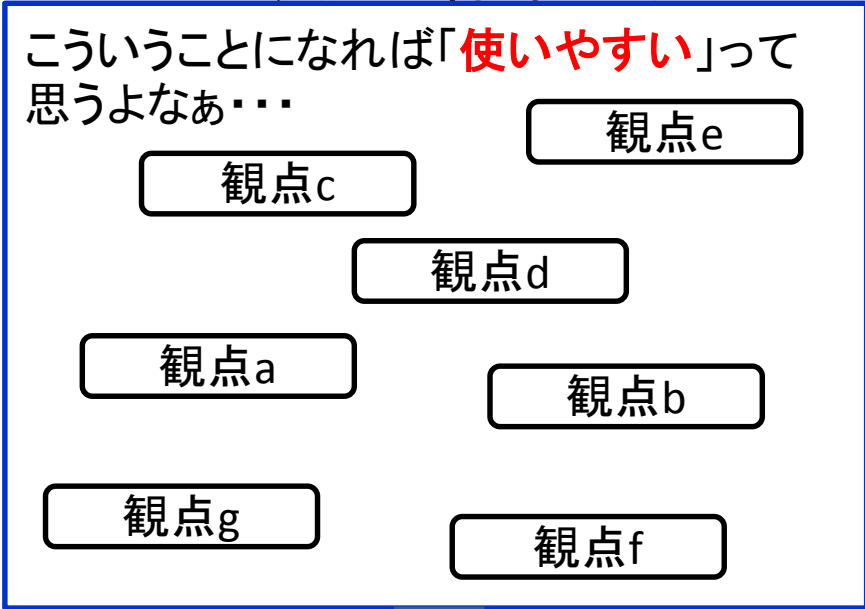
最上位観点を洗い出し、抜けを防止

関係者それぞれの立ち位置(立場)で、レビュー対象(この例では「要求定義・要求仕様書」)に実現してほしいコト、記載内容で確認したいこと＝“レビュー観点”を考え、洗い出す。



# レビュー観点の体系化(上下に行き来しながら)

## レビュー要求分析



# レビュー要求分析・設計・実装事例

## <前提条件>

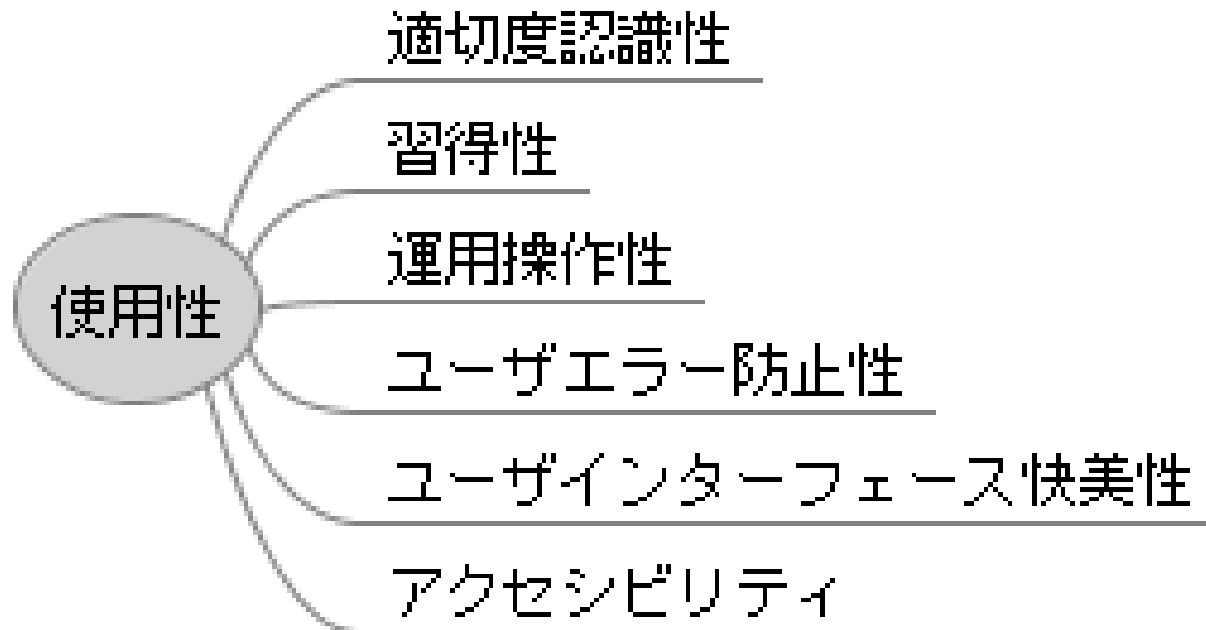
話をできるだけシンプルにするために、多くのシステム製品・観点の中からAmazon.comの「使いやすさ」で試行した。

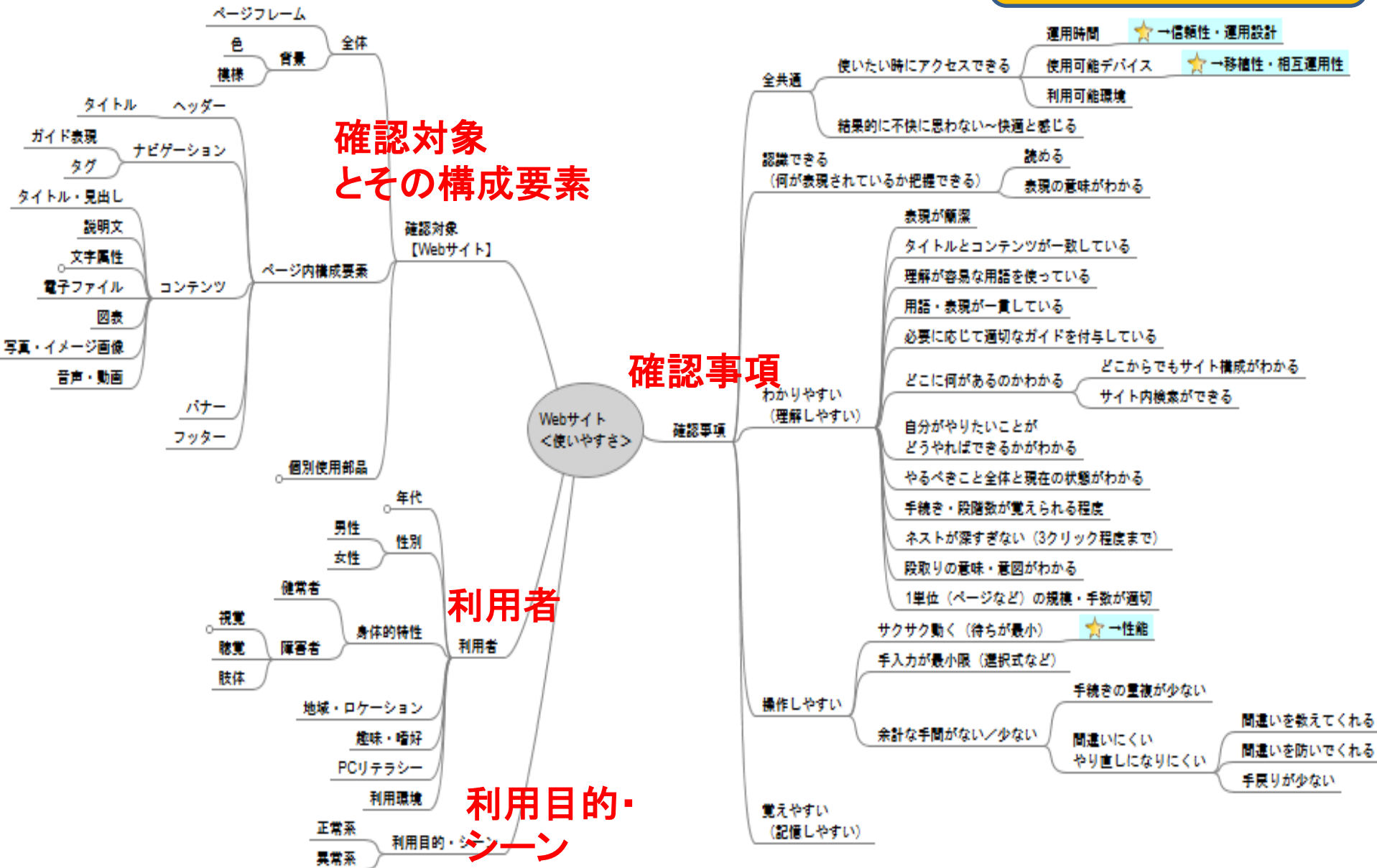
→要求定義フェーズの成果物(画面レイアウト・画面仕様・画面遷移など)に対するレビューを想定

汎用レビュー観点:使いやすさ(使用性)

レビュー要求分析

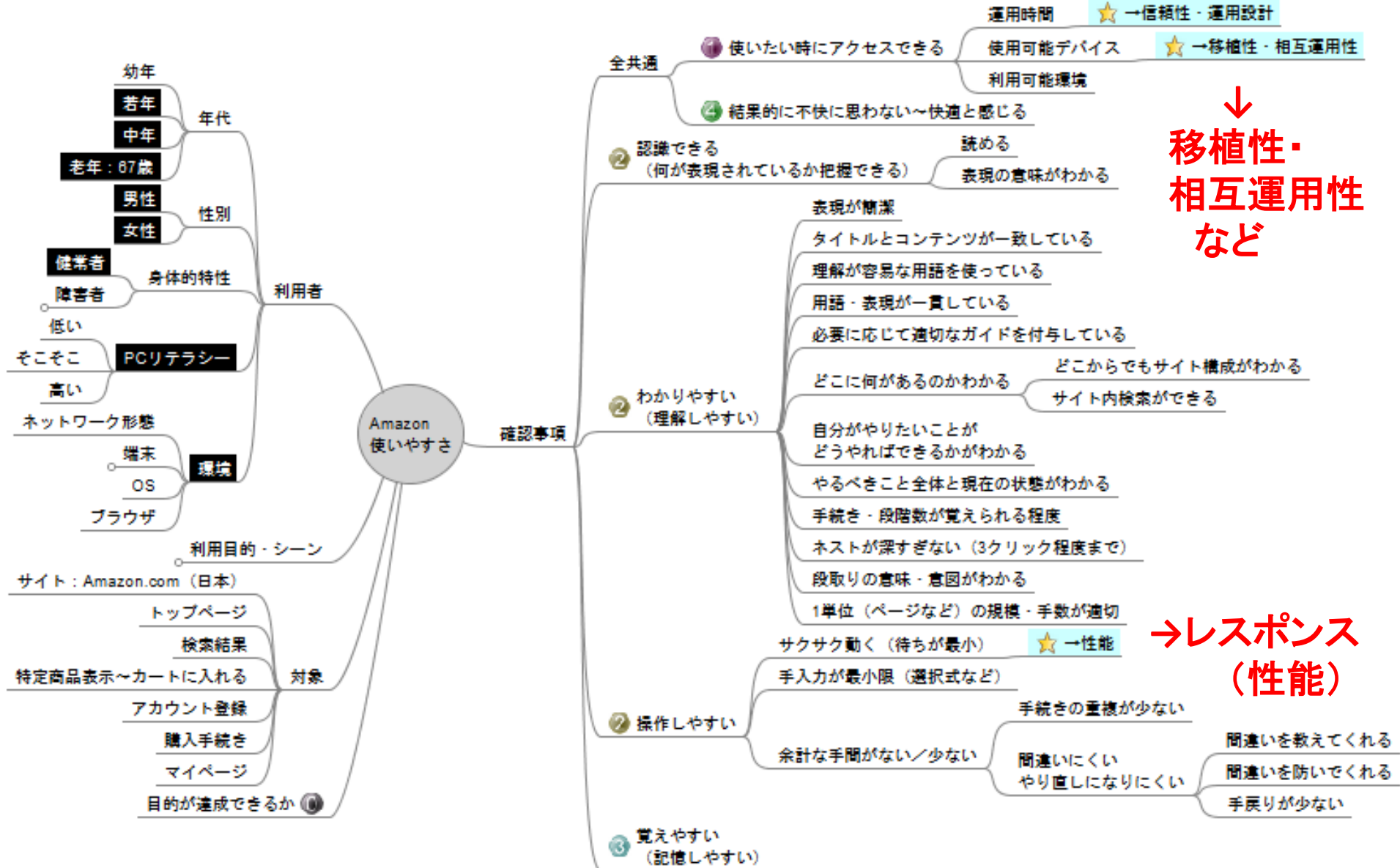
使いやすい  
だから...





# 製品特化型レビュー観点例～Amazon.com

## レビュー要求分析 ～設計

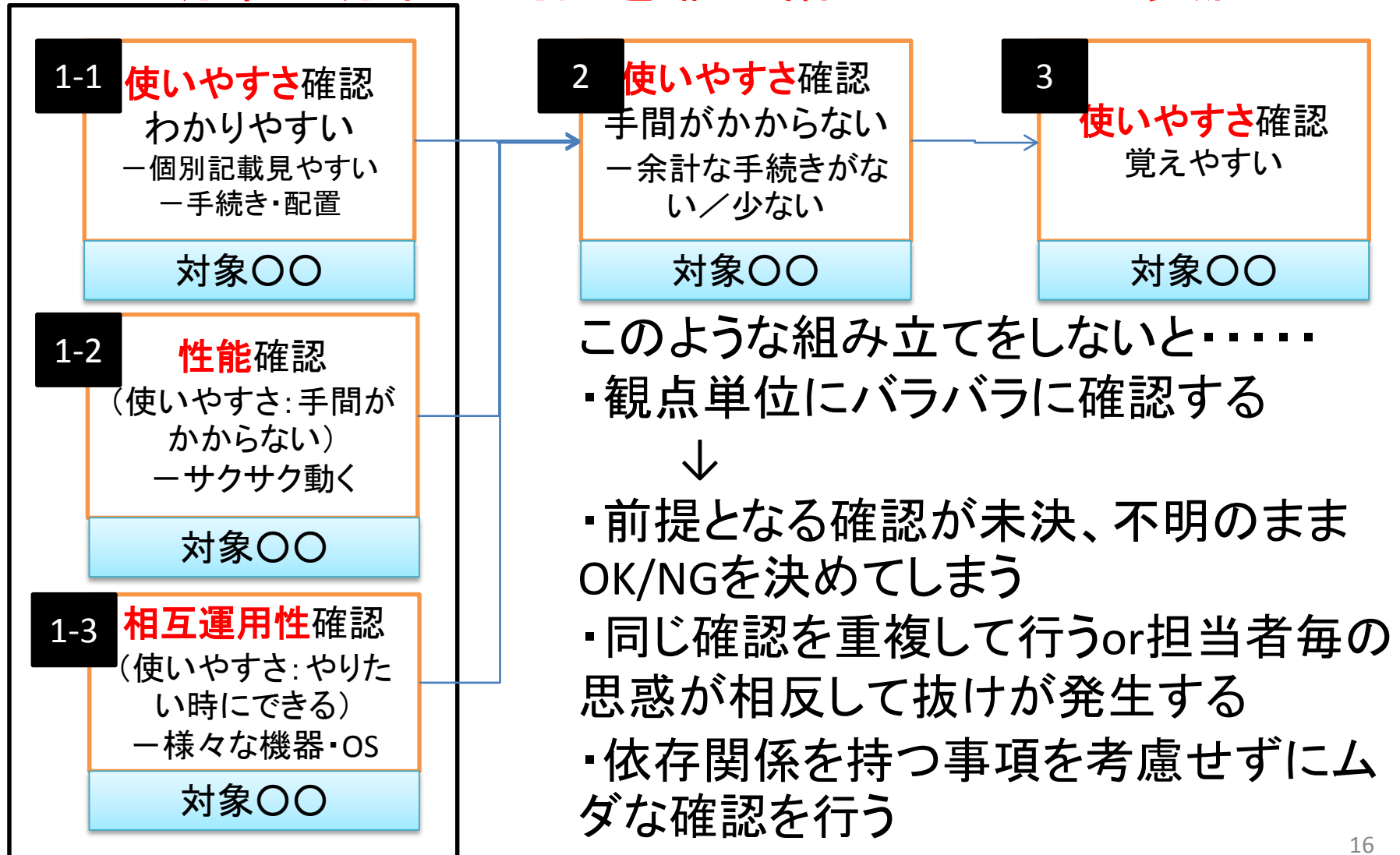


↓  
**移植性・相互運用性など**

→**レスポンス (性能)**

# 使いやすさを中心としたレビュー観点設計例

## →効果と効率の両面を兼ね備えたレビュー実施へ





# 1-1 レビューケース例

## そのまま確認して結果を判断できるレベル

### 具体的利用者とシーン



### 確認対象特定



### トップページ



### 確認対象特定

用語・表現が一貫している (統一感・整理されている)  
1単位の規模・手数が適切  
下部に思わない/楽しく使える

使いたい時にアクセスできる  
自分がやりたいことがどうやればできるかがわかる (そしてできる)  
やるべきこと全体と現在の状態がわかる  
間違いを防いでくれる  
間違いを教えてくれる  
間違っても最小限の手戻りで済む

自分がやりたいことがどうやればできるかがわかる (そしてできる)  
やるべきこと全体と現在の状態がわかる  
間違いを防いでくれる  
間違いを教えてくれる  
間違っても最小限の手戻りで済む

文字・写真・イメージが見やすい  
書いてある意味・意図がわかる  
どこに何があるのかわかる  
何ができるか予測できる (期待通りに動く)  
段取りがわかりやすい

文字・写真・イメージが見やすい  
書いてある意味・意図がわかる  
どこに何があるのかわかる  
何ができるか予測できる (期待通りに動く)  
段取りがわかりやすい

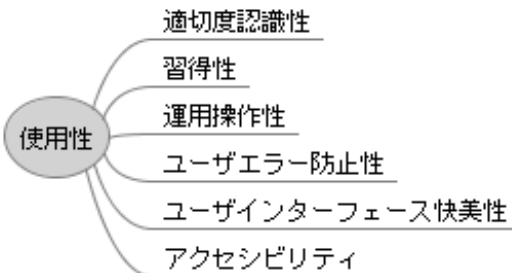
# レビュー要求分析～実装まで 実践してみてもわかったこと

- **一度に完璧なものはいできない。**
  - － 一人で行うのではなく、他者の意見、参考情報なども活用して可能な限りMECEにしていく
  - － 実活用結果からのフィードバックにより過不足、表現方法、構造・位置づけなどに対する見直しを行い、より活用しやすく効果があるものに育てていくことが必要
- **レビュー観点、確認項目の抜け・漏れ防止への効果が期待できる。**
- **併せて、レビュー実施の効率化が期待できる。**
  - － 論理的な観点単位にバラバラに確認する必要がなくなる
  - － 確認対象が小さく、確認項目が具体的なので確認と判定が楽にできる
  - － 関連する項目を連携しながら系統だって確認できる
  - － 重複している項目が集約できる
  - － 対象毎に不必要な項目を省くことができる
- **テスト要求分析(の多くの部分)とテスト設計の一部を兼ねることができる。**
  - － レビュー要求分析、設計時に、テスト要求分析、設計を並行で行うと効果的・効率的
- **レビュー対象成果物作成作業(今回の例では要求定義)の一部を兼ねることができる。**
  - － これができると、成果物案の質が高まり、レビュー時に確認する観点・項目をリスクが高いものに絞り込むことが可能となる
- **一方で、レビュー要求分析～実装まで手間と時間がかかる。**
  - － 実践力がつけば多くの部分で打開できると想定される
  - － さらに効率化するためには、レビュー観点図(スライド15・16)などの運営基盤となる情報を育てていくなどの対策が必要

# レビュー観点(設計)の活用と育成・進化

## 【観点ベース】

今回の事例では  
“使いやすさ”



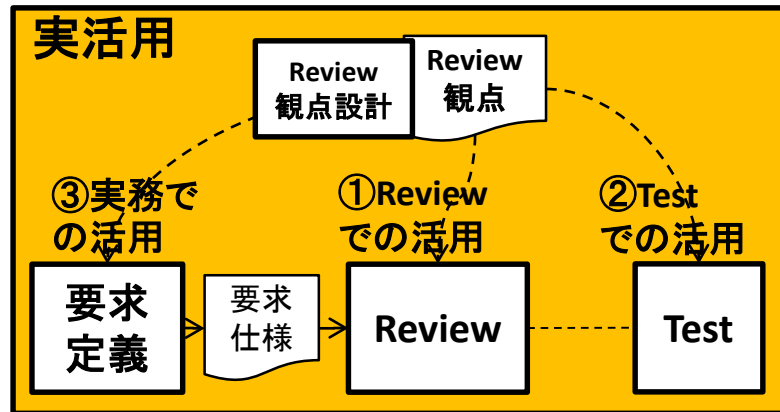
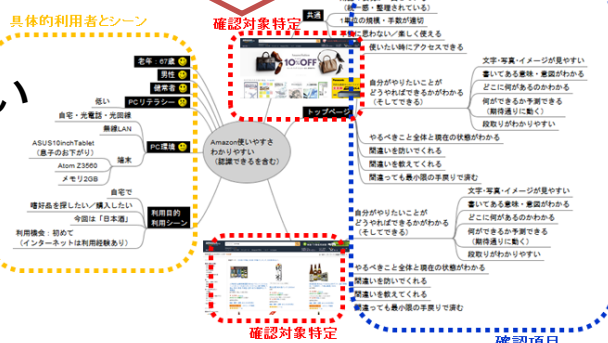
## 【特定分野】

今回の事例では  
“Webサイト・アプリ”の  
使いやすさ



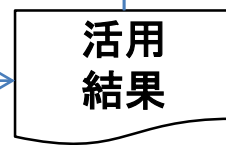
## 【特定システム】

今回の事例では  
“Amazon.com”の使い  
やすさ



Feedback & Update

Feedback & Update



# 参考文献

- SS2015 WG6 Position Paper
  - レビューの会議術からの脱却 西康晴さん
- JaSST'16東京 発表事例
  - レビュー目的・観点設定の効果と課題 安達賢二