2016.12.16　ソフトウェアテストシンポジウム2016四国　ワークショップ

【解説資料】

なんとなく実施する レビューからの脱却

～目的・観点設定に基づくレビューの効果を体感してみよう！

株式会社 HBA
Software Quasol 安達賢二
http://www.software-quasol.com/
adachi@hba.co.jp

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビュープロセスの全体像と今回の対象領域

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

目的・観点設定

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビュー目的・観点設定時に考慮すべきこと【背景の把握】の例

品質方針→品質計画→レビュー方針・戦略
→SCOPE決定
該当フェーズの目的
レビュー対象物の役割
プロジェクトのリスクと現状
プロダクトリスク
制約条件
入力情報とその状態
担当者のスキルと状況　など

目的は重要だが、まとめるのは簡単ではない

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

現在の立ち位置と関係するタスク・立場

例：要求定義フェーズ成果物案へのレビューの場合

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

立ち位置からの観点導出

- それぞれの立ち位置（立場）で“レビュー観点”を導出してください。

※ヒント：立ち位置で、「要求定義・要求仕様書」に実現してほしいコト、
「要求定義・要求仕様書」の内容で確認したいことを考えてみる。

- 導出した観点が複数ある場合は、重要度を1（最重要）から順に付与してください。

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

観点の体系化→目的の明確化

最上位の観点
ーレビュー目的群
※立ち位置から導出されることが多い

主に
マネージャ・リーダ
による具体化

主に
レビューア
による具体化

- 観点1
- 観点2
- 観点n
- 観点11
- 観点12
- 観点13
- 観点21
- 観点22
- 観点111
- 観点112
- 観点131
- 観点132
- 観点133
- 観点221
- 観点222

摘要

- 観点：当初自ら導出した観点
- 観点：自ら導出した観点を頼りに後から導出した観点

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

観点と粒度

トップダウンとボトムアップを組み合わせて
MECEな階層構造を構築する。
抽象度の高い領域で整理することで仕様変更
への柔軟な対応も可能になる。

段階的に詳細化し、粒度をレビューアが理解できるところに合わせることで、各レビューアが漏れなく確認し、判断することが可能になる。

例："使いやすさ"を具体化する過程

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

6W2H分析（事例）

※当事例は実利用者のみを対象としていますが、実際には、開発・販売組織なども対象となります。

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

プロダクト価値の例（利用状況分析）

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

プロダクトリスク評価→レビュー対象特定例

安全性構成要素	安全性を妨げる事象と想定される原因				対象 Hard	対象 Soft	対象 取説
健康への悪影響	やけど ※全体の95%以上	熱湯/蒸気流出・噴出・吐出		給湯などの誤操作・誤動作	ボタン・ポンプ	給湯制御	―
				ロック機能の動作不備	給湯ロック	ロック制御	―
				ポット洗浄剤等混入物	貯水槽	―	○
				勢いよく蓋を開閉	蓋	―	○
				転倒・傾斜	筐体	―	○
	けが	感電		漏電・短絡	筐体+電源コード	―	○
		転倒・落下など		コンセントひっかけ		―	○
	水質汚染	不適切な部材の使用		重金属などによる水質汚染	貯水槽	―	―
環境への悪影響	電気浪費	無駄な加熱・電力浪費		水温制御不備	ヒータ	水温管理	―
		不適切な部品・部材		部品・部材特性（性能）	筐体	―	○
経済的損失		漏電・短絡		漏電・短絡	電線周辺	―	○
	出火・火災	漏電		漏電・短絡		―	○
		短絡		コンセント異物付着	コンセント関連	―	○
	筐体・部品破損	転倒・落下など		コンセントひっかけ等×		―	○
				壊れやすい部材・部品	筐体	―	○

※参考：安全性構成要素情報：「電気ジャーポットの安全」 国民生活センター

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

表現形式変換による確認（例）

同一機能記述の集約・再編集による確認

状態遷移図の作成・追跡による確認

※シナリオの併用も効果的

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビュー・テストのための機能一覧作成（変換）事例

■話題沸騰ポット要求仕様書コンテンツ

http://www.sessame.jp/workinggroup/WorkingGroup2/POT_Specification_v7.PDF

1 ハードウェア構成と要求仕様
- 1.1 外観と内部構成
- 1.2 操作パネル部

2 操作要求仕様
- 2.1 コンセント
- 2.2 蓋
- 2.3 沸騰ボタン
- 2.4 保温設定ボタン
- 2.5 解除ボタン
- 2.6 給湯ボタン
- 2.7 タイマボタン
- 2.8 水位メータ

3 温度制御行為
- 3.1 沸騰行為
- 3.2 保温行為
- 3.3 温度制御行為をしない（アイドル）

4 温度制御方式

5 エラー検知

6 状態遷移
- 6.1 話題沸騰ポット状態遷移図
- 6.2 話題沸騰ポット状態遷移表

7 イベント

■話題沸騰ポット：テスト機能一覧

3 電源管理
コンセントon・コンセントoff

4 温度制御
- 4.1 蓋
  開ける・閉める
- 4.2 アイドル
  温度制御停止・アイドル中止
  操作パネル表示
- 4.3 沸騰機能
  沸騰開始・温度制御方式
  カルキ抜き・沸騰行為中止・停止
  操作パネル表示
- 4.4 保温機能
  モード切替
  保温開始・温度制御方式
  保温行為中止・停止
  操作パネル表示
- 4.5 水温管理

5 給湯機能
給湯開始・給湯停止
ロック・ロック解除

6 タイマ機能
タイマ開始・停止・リセット

7 水量管理

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

表現形式変換例（ガイド活用）

レビュー対象から機能記述を抜き出し、以下の枠に格納しながら確認する

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
レビュー観点例		
□入力元がわかるか？ □判定、処理に必要な情報（値など）やその範囲が記述されているか？ □参照情報の存在場所は記述されているか？	□判定方法・処理方法が記述されているか？	□出力先がわかるか？ □出力情報（値など）が記述されているか？ □出力条件やタイミングが記述されているか？

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

ラルフチャートによるポットをめぐる因子・水準の整理

（参考）詳説ラルフチャート（秋山浩一氏）
www.hayst.com/Documents/RalphChart.pdf

ノイズ

- **●電圧：瞬電・停電・不安定供給**
- ●電気周波数50Hz/60Hz
- **●気温／熱・気圧・湿度**
- **●傾斜地、机の淵、振動するところ**

アクティブノイズ

- **●ボタン長押し／連打**
- **●蓋半開き／強く締める**
- **●斜めにする、転倒させる**
- **△複数ボタン押し（同時・時間差）**

入力

- ●コンセント入れる／抜く
- ●蓋締める／開ける
- ●沸騰ボタン
  100msec以上押下

沸騰機能
with カルキ抜き

100℃まで沸騰 → +3分間カルキ抜き

出力

- ●沸騰ランプ点灯
  ※カルキ抜き終了時消灯
- ●保温ランプ消灯
  ※カルキ抜き終了時点灯
- ●水温表示
- ●水位点灯・点滅／消灯
- ●beep音：100msec
- ●カルキ抜き済100℃のお湯

状態

- ・アイドル・沸騰行為・保温行為・給湯・エラー
- ・ロック／ロック解除
- ・保温行為モード：高温・節約・ミルク
- **・水位変化（例：傾けるなど）**

湯沸かし時間の測定（性能テスト）が必要
→HW中心だがSW制御が適切かのテストにも該当

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

誤認識を引き起こす表現／対称漏れを見つける
ガイドワードによるレビュー

＜ガイドワード例＞

- 類似の意味を持つ複数の言葉
  例：完了・終了　警報・警告
- 複数の意味を持つ一つの言葉
  例：電源オフ・スタンバイ
- 非対称な機能遷移　<：上位遷移　>：下位遷移
- 条件指定：～のとき、～でない時、～の際、～場合
- 否定表現：～しない、～ではない
- 紛らわしい：詳細度が低い

＜効果的実践法＞
□ツールで検出する
　→そして判定・処理へ
□発生した不具合の内容を継続して蓄積・整理し、以降のレビューで活用する

出典：
鈴木三紀夫氏
意地悪漢字

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

シナリオ（事例）
ソリューション構想とレビュー、システムテストで活用される手法

①典型的利用者像

安達賢二　50歳
北海道生まれ
札幌の東隣江別市在住

1歳年上の妻、長男(24)・長女(19)の4人家族。
JR札幌駅前通りに面したITサービス企業に28年勤務している。
仕事柄、年中忙しく、家族旅行に行ったことがなかったが、勤務先の50周年記念で家族旅行がプレゼントされたのをきっかけに、11/3～5の日程で初めて四国旅行に行くこととなった。子供は都合がつかず妻と二人で水入らずの旅行である。北海道というお土地柄で自家用車にカーナビはなく、使ったこともない。初めて行く土地のため、航空機で新千歳→羽田経由→徳島入りし、レンタカー（これも初利用）のカーナビを頼りに徳島→鳴門→高松→こんぴら温泉→松山・道後温泉→しまなみ海道経由→宮島（厳島神社）を旅行する。

②実利用シナリオ

1：13:00に快晴19℃の徳島空港に到着し、徒歩3分のTレンタカーを借りた。車種は紺色4DoorのVits。13:30までに諸手続きが終了。
2：まずは最初の行先である「鳴門公園」に行きたいと思い、カーナビZの画面を確認し、「地名で探す」ボタンを押した。
3：入力画面で「なるとこうえん」と入力し「変換」ボタンを押すと、“鳴門公園（○○県）”と全国に存在する鳴門公園が北から順に多数列挙された画面となった。
4：“鳴門公園（徳島県）”が表示されるまでスクロールしてボタンを押すと、“ルート検索中です。しばらくお待ちください。”の表示となり、数秒後に高速利用、一般道利用、走行距離・走行時間などを表示した4種類のルートが表示された。
5：距離は15Km程度なので高速を使うまでもないと判断し、一般道を使う最も時間のかからないルートのボタンを押した。「ナビゲーションを開始します。まずは右折して道なりにしばらく走ります。」と音声が流れたので車を発進させた。
6：音声ガイドに従い運転し、途中コンビニに立ち寄り飲み物などを購入。
7：14:03に「目的地周辺です。」と音声ガイドが流れて鳴門公園駐車場に到着することができた。
8：駐車場に止めると、「目的地に到着しました。お疲れ様でした。」の音声ガイドが流れた。

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

何をどこまで記述するのか？

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

IEEE830：1998
ソフトウェア要求仕様書構成

1.はじめに

1.1目的　1.2範囲　1.3用語　1.4参考文献　1.5概要

2.概要・全体共通

2.1背景　2.2製品機能　2.3ユーザ特性　2.4制約
2.5仮定・依存性

3.詳細

3.1外部I/F　3.2機能　3.3性能要求　3.4論理DB要求
3.5設計制約　3.6ソフトウェア特性

4.索引

参考　IEEE830:1998

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

IEEE830：1998
ソフトウェア要求仕様が持つべき特性

正当性(Correct)：
持つべきすべての要求がふくまれ、以外の要求をふくまないこと

無あいまい性(Unambiguous)：
すべ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
すべての必要な要求
すべての入力データと状況に関する応答の定義
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て検証できること

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

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

プロダクトに求められる特性の例

■利用時の品質：システムやソフトウェアの利用とその結果に着目した特性

■システム／ソフトウェア製品品質

システムやソフトウェアそのものの特徴を表す要因を分類したもの

出典：JIS X25010：2013より

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

欠陥を狙い撃つ！

リソースは限られている！

⇒欠陥がありそうなところ、怪しいところを狙い撃つことが重要。

- 条件（分岐）があるところ／条件の切り方＋境界値の周辺
- 込み入っているところ／複雑なところ／混雑しているところ
- 曖昧に記載されているところ
- 対称語が存在するのに記載がないところ
- 抽象用語で記載されているところ
- 別の場所にも同じような表現があるところ
- 誤字や脱字が多い対象物
- 変更が多発したところ／急に決まったところ
- 時間が不足していたところ／駆け込みで作成したところ
- スキルや知識が不足しているところ、不得意なところ
- これまでに欠陥が多発しているところ
- 要求や仕様、担当者、組織が変わったところ
- 分かるような、分からないようなところ／複数の受け取り方ができるところ

ソフトウェアテストの一般原則

出典：ISTQBテスト技術者資格制度 Foundation Levelシラバス日本語版 Version 2011.J02

原則4：欠陥の偏在

リリース前のテストで見つかる欠陥や運用の故障の大部分は、ある特定の少数のモジュールに集中する。

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

検出効果の高い欠陥・リスクとは？

この先、この欠陥がそのまま残留したと仮定した場合、処置・改修に要する工数・期間が大きいもの。
あるいは、プロジェクトを危機的にする、顧客やエンドユーザに損害を与える度合いが大きいもの。

一般的には、「現時点」から「発見されると想定できる時期」までの間隔が長いほど、処置・改修にかかる工数は膨らんでいく。

ただし、中にはいつ発見されても処置・改修にかかる工数が変わらない欠陥・リスクもあるので注意。

→意味を取り違うことがない誤字・脱字・衍字などはその代表例

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

要求仕様書でよくある欠陥の例

- 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
- 優先度未設定　など

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビュー実践ワーク（応用編）結果（とある組織の例）

＜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	技術的実現性の課題把握	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
想定リスク	●●	△△	●△	□■	■●	○△

※当ワークの結果から、その組織が普段何を重視してプロダクトを開発しているかとその一貫性（徹底度合）、レビュー実践内容が把握できる。

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

要求仕様書レビュー汎用的観点と確認手段例

No	レビュー項目	概要	確認手段例
1	合目的性	システム化により解決したいことが明確であり、記述内容（システムに求められる事項）により解決できるか	システム目的記述があるか 記述内容（例：機能）により目的は達成可能か
2	記述事項網羅性	記述すべきことが漏れなく書かれているか	システム目的・要求（機能・非機能）・仕様が記載されているか
3	明確さ・分かりやすさ	記述されている内容の意味が明確か／理解しやすいか	文章ばかりになっていないか 図表等で分かりやすく表現されているか
4	追跡可能性・容易性	記述内容の起源が明確で、追跡可能か／追跡が容易か	記述内容の起源が追跡できるか／識別子があるか
5	整合性（内部・外部）	記述内容と関係する箇所（内部・外部）との間に矛盾はないか	同左
6	優先度・重要度明確性	優先度・重要度の高い機能等とそれ以外が明確になっているか	優先度・重要度の判断基準と評価結果／順位が設定されているか

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

マインドマップによるレビュー観点設定事例

「見どころ」＝レビュー目的
ウリを備えた利用者の課題解決製品として成立する
要件とSPECを備えているかを確認する

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

「アドホック」→「目的と観点設定」の変化による改善効果事例

改善前検出△ 改善後検出●			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 など	5	●●● △		●●●●●●●●●●●●● △	●●	△30 ●335
	効果中 主対象： 機能上のバグ など	3	● △△ △△△ △△△	△△	●●●●●●●●● △	△	△78 ●93
	効果小 主対象： 規約違反 誤字・脱字・衍字 など	1	● △△△ △△△ △△△			● △	△16 ●8
計			△38 ●19	△18 ●0	△40 ●340	△28 ●77	△124 ●436

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

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

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

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

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

指摘価値変化例

指摘件数: △23 → ●24

△改善前(Before) ●改善後(After)			発見可能Phase(想定)				計
			設計～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 余計な要件・仕様	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

△が左下
(指摘価値が低い)に集中

●が右上
(指摘価値が高い)
に集中

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

結果が出なかったチームの例

指摘件数: △14 → ●11

△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 ハード変更必要 サーミスタ壊れたら止められない ミルク温度検出仕様なし	5		●	●		0 ↓ 40
	効果中 沸点記述 整合性 デフォルト エラー後復帰×	3		△ ●●	△△△△		69 ↓ 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 ↓ 7
計			9→7	9→33	60→25	0→0	78→65

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

効果が出なかった理由（考察結果）

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

結果評価・判定とふりかえり

ピアレビューの費用対効果は、バグがどれだけ見つかったかではなく、プロジェクトチームの作業時間がどれだけ節約できたかで判断すべきです。

～書籍「ピアレビュー」Karl E.Wiegers

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビュー指摘事項の評価

No.	指摘箇所	指摘内容	指摘分類
①	筐体	筐体と構造が“手軽”でないのでは？	ウリ・使用性
②	蓋	閉めると沸騰モードになるのはわかりにくい。	ウリ・使用性・安全性
③	タイマ	単独タイマ機能でコストが高いなら必要ないのでは？（100均ショップで分秒表示タイマ購入可能）	有効性・効率性
④	カルキ抜き	蒸気を逃がす機構なし。カルキが抜けないのでは？	機能性
⑤	解除ボタン	100msec押下で解除可能。子供のいたずら・偶然の押下でも解除できてしまう可能性あり。	安全性
⑥	解除ボタン	ボタン表示「解除」では何の解除かわからない。	ウリ・使用性
⑦	アイドル移行	pot-330-12・13では蓋センサoffが条件となっているが、pot-221-11では蓋センサが1sec以上offとなったら温度制御行為中止となっている。不整合では？	不整合
⑧	ミルクモード	対象ユーザのニーズなしでは？必要ないのでは？	有効性
⑨	ミルクモード	沸騰・カルキ抜き後から60℃になるまで時間がかかりすぎないか？（電源on時は60℃に到達しない？）	効率性・有効性
⑩	コンセント	筐体との接続部分が安全なのかわからない。	安全性

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビュー効果分析表

			今回見逃した場合の発見可能Phase（想定）					計
			設計	実装・UT	IT	ST・OT	C/O後	
			1	3	5	7	9	
観点効果／見逃し後影響度	【効果大】 検出欠陥概要 余計な要件・機能 要件・機能漏れ 安全性不備	5						
	【効果中】 検出欠陥概要 機能上のバグ 設計に影響有の不整合 重要度小の非機能不備	3						
	【効果小】 検出欠陥概要 設計に影響小の不整合 規約違反	1						
計								／

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビュー評価結果例　改善効果の実感が次の改善へのモチベーション！

▲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 機能仕様漏れ	5	▲▲ ●		▲ ●●	●●●	35 ↓ 160
	効果中 設計に影響あるもの	3	▲▲▲ ●●●●●		▲ ●●●●● ●●●		24 ↓ 135
	効果小 誤字脱字	1	▲▲ ●●		●●		2 ↓ 12
計			21→22	0→0	40→180	0→105	61→307

レビュー観点設定を改善することにより、（アドホックにレビューよりも）指摘内容がどのように変化したかを比較・評価した例

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビュー結果の簡易判定

結果判定

□合格　□条件付き合格　□不合格

□指摘事項の修正後、再度レビューを実施する。
□指摘事項の修正状況と修正結果をモデレータが個別にフォローアップする。
□指摘事項の修正結果は、担当者自らが確認し、終了したことをモデレータに報告する。
□指摘事項の修正結果は、自ら確認し、問題なければそのまま次のフェーズに進む。

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビューのふりかえり

レビュー終了後その場で、短時間でふりかえる
すぐやると効果大／後に回すと記憶が・・・効果薄

→レビュー対象の品質、レビュー目的の達成／未達成、レビュー運営の良い面・改善が必要な面（今回の改善の成果を含む）をその場で確認
役割毎のパフォーマンス、レビュー観点・確認方法の適切性、基準値と採用方法・条件による調整方法等も考慮
次にどうするのかまで具体的に取り決めておく

→欠陥内容～作業方法の改善をすり合わせる
作業のインプット情報活用の仕方、作業項目設定、段取りの組み方、レビュー運営設計の良し悪し・・・など
次に手戻り少なく、欠陥の少ない成果物案を出せるように

メトリクスなしでも相当なパフォーマンス改善が可能

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビューふりかえり

- 「ふりかえり表」を活用して今回のレビューをふりかえる。

(1)「今回やったこと」を記載する。（任意）

(2)（(1)の結果から）主要な観点に沿って今回わかったことを書き出す。

(3)(2)の結果から、実践すると効果が得られる「次にやること」を具体化する。

(4)関係者で内容を共有し、納得する。

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

ふりかえり結果例

Y:やったこと＆結果	W:わかったこと	T：次にやること
◆やったこと◆ ・モデレータ1名＋レビューア3名 ・それぞれに観点割り付け→個別レビュー後に集合レビュー ・集合レビューではウォークスルーを採用 ・一度レビュー終了後、追加観点で再レビュー実施	Keep：実践・継続したいこと 〇ウォークスルーで作成者が説明すると、著者の意図と記述内容の差異が明確になり、指摘事項を的確に伝達しやすい。 〇レビューア毎に観点を割り付けて実施すると検索範囲を絞りやすく、集中できるのでよい！ 〇Aさんのモデレート、神ってる！みんなを前向きにしつつ、コメントを引き出していた！ 〇余計な機能や基本構造の不備等が指摘できたので費用対効果がすごい！	□Aさんのモデレートのよいところをまとめる ・アイスブレイクから入る ・目的、注意事項共有 ・笑顔でハキハキ伝える ・自分が対象の内容を理解する過程をみなで共有するなど →次のモデレータが自分でできるところから真似て進めてみる。
◆結果◆ ・欠陥指摘10件 重大5・通常3・軽微2 ・速度　500L/h ・欠陥密度　8件/KL ・費用対効果80!!	Problem：問題・課題事項 ×観点がピンポイントに偏ったかも。指摘事項にムラがある＆欠陥密度が低い。 →仕様・機能詳細などモノづくりの視点の指摘が少ない→追加レビュー実施 ×1分間に8.3行。レビュー速度、速いんじゃないか？	□観点設定結果をモデル図で示し、有識者と簡単に抜け・漏れ確認をしてから割り振る。 □試行：パラグラフ単位にレビュー速度（5行/分程度まで）を確認しながら進めてみる。→その結果何が変わるか評価する。

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビューの効果は下からの積み上げ

狙った効果を獲得できる

誰もが効果を納得できる

適切な目的・観点を設定できる

効果を客観的に評価できる

適切な目的・観点で実践できる

効果を実感・共有できる

必要な情報が事前に準備・入手できる

各種基準・タイプ・技法を使いこなせる

指摘事項をチーム内で共有できる

（提示→受取・納得（→修正）が自然と前向きに行われる場）

Copyright © Kenji Adachi@HBA Quasol, All Rights Reserved

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。

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

運営面の役割は「指摘しやすい環境づくり」

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビュープロセスの全体像と今回の対象領域

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

最上位観点のバラつき

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

観点設定がなければ、レビューで確認されない

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

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

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

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。

×固定な手続き適用・画一的な回答獲得
○思考を伴う段階的詳細化／構成要素による構造化

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビュー要求分析

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。

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

追加資料

現在の立ち位置と関係するタスク・立場

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

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

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

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・・・

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

製品領域汎用レビュー観点例～Webサイト／アプリ使いやすさ

レビュー要求分析～設計

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

製品特化型レビュー観点例～Amazon.com

レビュー要求分析～設計

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビュー設計

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 使いやすさ確認
わかりやすい
－個別記載見やすい
－手続き・配置
対象〇〇

1-2 性能確認
（使いやすさ：手間がかからない）
－サクサク動く
対象〇〇

1-3 相互運用性確認
（使いやすさ：やりたい時にできる）
－様々な機器・OS
対象〇〇

2 使いやすさ確認
手間がかからない
－余計な手続きがない／少ない
対象〇〇

3 使いやすさ確認
覚えやすい
対象〇〇

このような組み立てをしないと・・・・・
・観点単位にバラバラに確認する
↓
・前提となる確認が未決、不明のままOK/NGを決めてしまう
・同じ確認を重複して行うor担当者毎の思惑が相反して抜けが発生する
・依存関係を持つ事項を考慮せずにムダな確認を行う

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビュー実装

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

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビュー要求分析～実装まで実践してわかったこと

＜指摘件数・内容のばらつきを低減できる（可能性大）＞

- **一度に完璧なものはできない。**
  - 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ェ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、レビュー観点の重複防止→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

57

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

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

58

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

前フェーズ実践方法と成果物の状態により レビューの内容や作業負荷が変動する

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

レビュー実践内容のSCOPE

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved

ぜひ実務で活用してみてください！

（実践⇒評価・改善）×3回が理想です

さらにレビューを深掘りしたい方はこちらを参考にしてください。

	リーダ 管理者	モデレータ	作成者 レビューア
レビュー計画～評価・ふりかえり	◎	○	△
レビューメトリクス	◎	○	△
レビュー改善～開発プロセス改善	◎	◎	○
組織的レビュー観点共有・活用	◎	◎	○
レビュー観点モデリング	○	◎	◎
ファシリテーション・モデレーション	○	◎	○
レビュー観点・目的設定による実践	○	◎	◎
レビュー基礎	○	◎	◎

今回はこの
ショートバージョン
を実践しました

Copyright © Kenji Adachi@HBA Quasol , All Rights Reserved