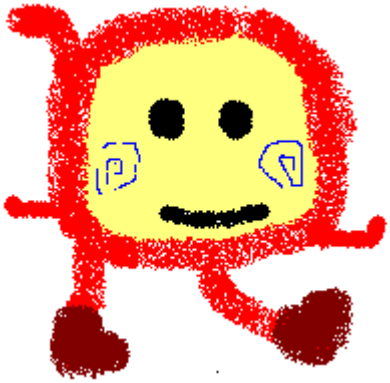
これからレビューを始める人、
レビューが上手くいかない人のための

レビュー・オリエンテーション・キット

レビューの基礎知識、レビューを成功させるための
ノウハウや心構えを紹介します。



Review Orientation Kit

はじめに

レビューは欠陥の速やかな検出および混入予防に有効な技術である。またレビューの適用範囲は設計書やソースコード、あるいはプロジェクトの進捗度合いや標準プロセス準拠の確認など広範囲に及ぶ。

欠陥の検出や予防の目的でレビューを導入および実施している組織は多くあるが、その全てで期待した効果が得られているわけではない。考えられる原因はレビューに関する知識不足、レビュー目的の不在、軽微な指摘内容、関係者の人間関係など様々である。

そこで、誰でもレビューによって欠陥検出や成果物の品質改善などに対して一定の効果を得られるよう、レビューの教育キットを作成することにした。

レビューをこれから始める人は勿論のこと、レビューを実施しているが期待している効果が得られていない人、レビュー実施することに対して懐疑的になっている人にも、本キットを読んでもらい、効率的・効果的なレビューを実施して頂きたい。

レビュー・オリエンテーション・キットの構成

第1章. レビュー・オリエンテーション・キットの紹介

本レビューオリエンテーションキットの概要を紹介する。

第2章. レビューの基礎

レビューとは何か？レビューは何のために行うのか、どんな利点があるのか、レビュー技法の種類と特徴、レビュー目的の決め方、レビュー観点の決め方、レビュー時の役割など、レビューの基礎を学ぶ。

第3章. レビューの演習

知識を学んでも、すぐに実践には活かせない。演習によるレビューの練習を行う。

第4章. プロジェクトへ導入する際のヒント

実際のプロジェクトへレビューを導入する際に、上手く導入するためのヒントを紹介する。

第5章. レビュー文化の醸成

レビューを導入しても、いつの間にか形式的になってしまう組織が少なくない。レビューを継続的、且つ、効果的に行い、レビュー文化を醸成していくためのヒントを紹介する。

別冊. レビュー標語録

レビューを実施するようになった中級者以上向け、レビューの実施や改善などにすぐに役立つノウハウ、心構え、標語を、別冊としてまとめた。

目次

第1章　レビュー・オリエンテーション・キットの紹介

本レビュー・オリエンテーション・キットの目的は「誰でもレビューが出来るようになること」である。このキットの内容を読み進めることで、レビューに関する基礎的な知識が身に付き、演習によって知識の確認とレビューの効果を体験し、開発プロジェクトへのレビューの導入と、レビューの継続によるスキルや品質の向上と言ったレビュー文化の醸成について学ぶことが出来る。

またレビュー経験者向けには別冊のレビュー標語録を参考にすることで、現在行われているレビューの改善に向けたノウハウを得ることが出来る。レビューの活動を改善し続けることで、自分自身のスキル向上、製品の品質向上、手戻りの防止、コミュニケーションの改善など様々なメリットが得られる。このキットの内容を読み進めると、レビューの活動にはレビュー以外の様々な知識も必要になることが理解できるので、レビューを実施できるようになったら次は長期的な目標を持って活動を進めていくことをお勧めする。

第2章　レビューの基礎

レビューの定義と効果、レビューの種類と特長、レビューの目的、レビューの観点、レビュー参加者の役割といったレビューの基本的な知識を紹介する。

2.1. レビューの定義と効果

（1）レビューの定義

レビューとは、「作成された仕様書やソースコードなどの成果物に対して、作成者とは別の人が調べて、欠陥を見つけ出してあげること」である。作成者以外の人が確認すると、作成者本人では気付かない文章の曖昧さや多義性、前提となっている条件自体の問題などの欠陥が検出しやすい。

このように成果物に混入した欠陥を検出するためのレビュー以外にも、「マネジメントレビュー」や「マイルストーンレビュー」などプロジェクトの継続や次工程への移行を経営層や責任者に承認してもらうためのレビューや、技術移転を目的とした教育のためのレビューなどがあるが、本キットでは「成果物に混入した欠陥を検出するためのレビュー」を対象とする。

品質保証の活動として、レビュー以外にテストがある。テストはソフトウェアを実際に動かして調べるのに対して、レビューはソフトウェアを動かす前に調べるという点がテストとの大きな違いである。

（2）レビューの効果

レビューの効果として、主に以下のようなものがある。

どの効果を狙うのか、戦略的にレビューを実施することが重要である。

・手戻り工数の大幅な削減

仕様書やソースコードが作成できたらすぐに実施できることと、欠陥を直接検出できることから、テストに比べて手戻りコストが大幅に削減できる。「4.4. レビューによるコスト削減効果」参照。

・品質に対する意識向上

自分が作成した成果物を人に見られるということから、低品質のものを見せたくないという心理が働き、より高い品質のものを作成しようという意識が高くなる。

・スキルアップ、知識移転

レビューで指摘を受けることで、欠陥に関する知識やより良い成果物を作成するための知識を習得することができる。またベテランから初心者への技術の移転にも繋がる。

・成果物の品質向上

一度指摘された欠陥を次は混入されないように努めるので、成果物の品質が向上する。

・認識の共有

成果物の内容についてレビュー参加者間の認識齟齬を無くすことができる。

レビューの効果を得るための工夫については、「第4章　レビューをプロジェクトに導入する際のヒント」、「第5章　レビュー文化の醸成」で解説する。

2.2. レビュー技法の種類と特徴（やり方・長所・短所）

レビュー技法には以下のようなものがある。

インスペクションが最も厳格で効果が高いが、レビューにかかるコストも高い。どの技法を選択するかは、プロジェクトの求められる品質レベルや、対象とする成果物に応じて、各技法の特徴を考慮して、取捨選択する。

「4.2. 開発プロセスへの技法マッピング」参照。

技法	やり方	長所	短所
アドホックレビュー	計画せずに、必要に応じて対応できるメンバで確認する即席レビュー。	目の前の問題解決に即応できる。 準備コストが掛からない。	自由度が高いため、欠陥や矛盾の指摘から話がそれてしまうことがある。
パスアラウンド	多重、同時進行型のチェック。成果物を複数人へ配布し、フィードバック（コメント）を依頼。	1人のレビューアのチェック漏れや、雑なチェックリストを予防できる。 レビューアの時間に制約がない。	「他人の意見をきっかけに発生するアイデア」といったレビュー活性化の刺激がない。 レビューアに責任がない場合、効果的に指摘が出ない。 類似コメントが多く発生し、レビューイが混乱する。 成果物の最新版管理がレビューアに依存する。
ペアレビュー（ピアデスクチェック）	作成者とレビューア（1人）がペアを組み成果物を確認。対面レビューの前にレビューアが机上試験（ピアデスクチェック）を行う場合もある。	技術の高いレビューアがいて、十分な時間を使える場合は非常に有効な手法。 1人分のレビュー時間のため、最も低コストのレビュー手法である。	完全にレビューア1人の知識と技術に依存するため、レビュー結果は多様なものになる。
ウォークスルー	作成者が成果物を複数人へ説明しコメントを求める。作成者の理解を共有し、考えを一致させることが目標。	説明を行うことで、作成者自身が今まで気付かなかった事項に気付く。 説明後の他人のコメントを参考に品質を改善することができる。 作成者の考えを共有する場合や、テストケースチェック等に有効。	作成者は説明に集中する傾向にあり。欠陥を見つける行為が鈍る。 一方的な説明会になりがち。（レビューアは、記載していない行間を個々に解釈してしまう） 説明を割愛した部分の欠陥は見つけにくい。 レビューアは説明された箇所から気付いた範囲の指摘になりがち。 結果が曖昧で終わる場合や、指摘反映がなおざりになる場合がある。
チームレビュー（一般的なレビュー）	複数人の適格なメンバで成果物を確認。計画／手順／基準に沿って実施。作成者が説明と共に会議進行も兼任することもある。	レビューが計画的に行われ、様々な参加メンバにより欠陥が摘出される。	レビューアの事前準備がない場合、一方的な説明会になる場合がある。 レビュー時間が、解決案の議論の場や実装方式のコンセンサス形成の場になることがある。
インスペクション	作成者以外の人が説明や会議進行を行い、作成者は指摘内容の理解に専念する。フォローアップも確実に行う最も体系的で厳格なレビュー。	レビューが計画的に行われ、欠陥の摘出を効果的に実施することができる。	他のレビュー手法に比べ、工数が掛かる。（事前準備と計画も含めて）

（参考文献[1]より）

2.3. レビュー目的の決め方

何のためにレビューを行うのか、目的を決めて、関係者で合意してから、レビューを実施する。普段は成果物の欠陥検出を目的とするが、初回は標準適合性に重点を置く、業務未経験者が作成した成果物を見るときは業務知識の移転も目的とするなど、プロジェクト、作成者、成果物などの状況に応じて目的を設定する。欠陥検出が目的の場合でも、軽微な欠陥を出来る限り取り除くことを目的とする場合と、軽微な欠陥は無視して重要な欠陥を検出することを目的とする場合では、検出される欠陥の種類も変わってくる。

ただ漠然とレビューするのではなく、参加者全員で共通の目的意識を持ってレビューを実施し、目的に沿わない議論になった場合は軌道修正を行いながらレビュー会議を進めることで効率的なレビューが実施できる。

以下にレビューの目的を決める方法についての例を紹介する。

1) 段階的欠陥除去

欠陥をいくつかのランクに分けてランクの低い欠陥から順番に検出する。

例えば以下のような順序で目的を変えながら段階的に実施する。

①文法エラーの除去を目的とする。静的解析ツールで検出可能な欠陥を検出する。

②日本語の曖昧な表現や多義性の排除を目的とする。初級レビューアに担当させる。

③他の文書との整合性確保を目的とする。初級～中級レビューアに担当させる。

④業務上問題がないことの確認を目的とする。有識者に見てもらう。

2) 不安解消(CDR 法)

各ステークホルダが不安に思っていることをヒアリングし、その不安を解消することを目的とした、レビュー観点を設定する。

(参考文献[2]より)

2.4. レビュー観点の決め方
(1)レビュー観点の設定方法

レビューの目的に応じて、レビュー観点を設定する。レビュー観点を設定することで、どのようなチェックを行えば良いかのヒントとなる。例えば、レビュー観点が「セキュリティ」であれば、「誰でもアクセスできるようになっていないか」などの確認項目が過去の経験や保有知識を元にして浮かんでくる。

ISO9126 の品質特性を用いて、どの品質特性について確認を行うか大まかに決めた後、各プロジェクトや成果物に応じた詳細な観点を設定することが多い。更に過去の欠陥情報があればその教訓を活かした内容にするのが一般的である。それ以外にもステークホルダーの視点に立った観点を設定する方法などもある。

以下にレビュー観点の例を紹介する。

1) 過去の欠陥情報

過去にレビューやテストで摘出した又は出荷後に発覚した欠陥を、障害データ、レビュー記録、テスト結果、チェックリストなどで確認し、類似欠陥が無いかという観点でレビューを行う。

2) ISO9126 品質特性

ISO9126 品質特性の6つの主特性(機能性、信頼性、使用性、効率性、保守性、移植性)、およびその副特性を用いてレビュー観点を設定する。

3) ステークホルダ

各ステークホルダの視点でレビューを行う。実際にそのステークホルダーにレビューへ参加してもらうのが困難な場合は、そのステークホルダーになったつもりでレビューを行う。（お客様、エンドユーザ、テスト担当者、保守担当者など）

4) 工程

後工程のことを考慮した内容になっているかという観点でレビューを行う。（実装可能か、テストし易いかなど）

5) 標準化

社内標準やプロジェクト内ルールに従っているかという観点でレビューを行う。（文書作成ルールなど）

（2）レビュー観点の絞込み

様々な観点でレビューを行う方がより多くの欠陥を検出できることは確かであるが、全てのレビュー観点でレビューすることは現実的には不可能である。どの観点でレビューするのか絞込みを行い、的を絞ってレビューを行う方が効果的である。例えば一度に全ての観点でレビューするのではなく、複数回に分けて毎回異なる観点でレビューした方が集中できるため検出効率が良い。レビューアが複数名の場合は、全員が同じ観点でレビューするのではなく、それぞれ異なる観点でレビューした方が効果的である。以下にレビュー観点を絞り込む方法を紹介する。

1) 間接的メトリクス

間接的メトリクスを用いて欠陥予測を行い、含まれていそうな欠陥を検出するためのレビュー観点を設定する。 「参考文献[3]より」

2) リスク

リスクが高いものにターゲットを当てる。（人命に関わる、業務が停止するなど）

3) テストケース

修正確認テスト規模が大きくなる欠陥を予想しながら優先的に検出する。

「参考文献[4]より」

2.5. レビュー時の役割

（1）レビュー時の役割

レビューに参加する人の役割を決めて、各自が役割を意識して望むことで効果的にレビューを進めることができる。参加者の役割としては、以下のような役割がある。全て専任とした方がより効果の高いレビューが実施できるが、参加者の人数が足りなければ、兼務すれば良い。

―「参考文献[1]より」

役割	実施事項
司会進行役（モデレータ）	・レビューをリードする ・相乗効果を刺激する ・レビュー結果を報告する
読み手	・記述を解釈して言い換えをする
検証者（レビューア）	・仕様の欠落や矛盾、懸念点を指摘する
作成者（レビューイ）	・質問に答える ・読み手の言い換えが意図した内容と合っているか確認する
記録者	・指摘を記録する

（2）レビューアの役割

レビューアが複数名の場合は、全員が同じ視点で確認するのではなく、各々別の観点から欠陥を検出するように役割を分担する方が、脇道にそれることなく効果的なレビューが実施できる。

レビューアの役割分担の例を以下に紹介する。

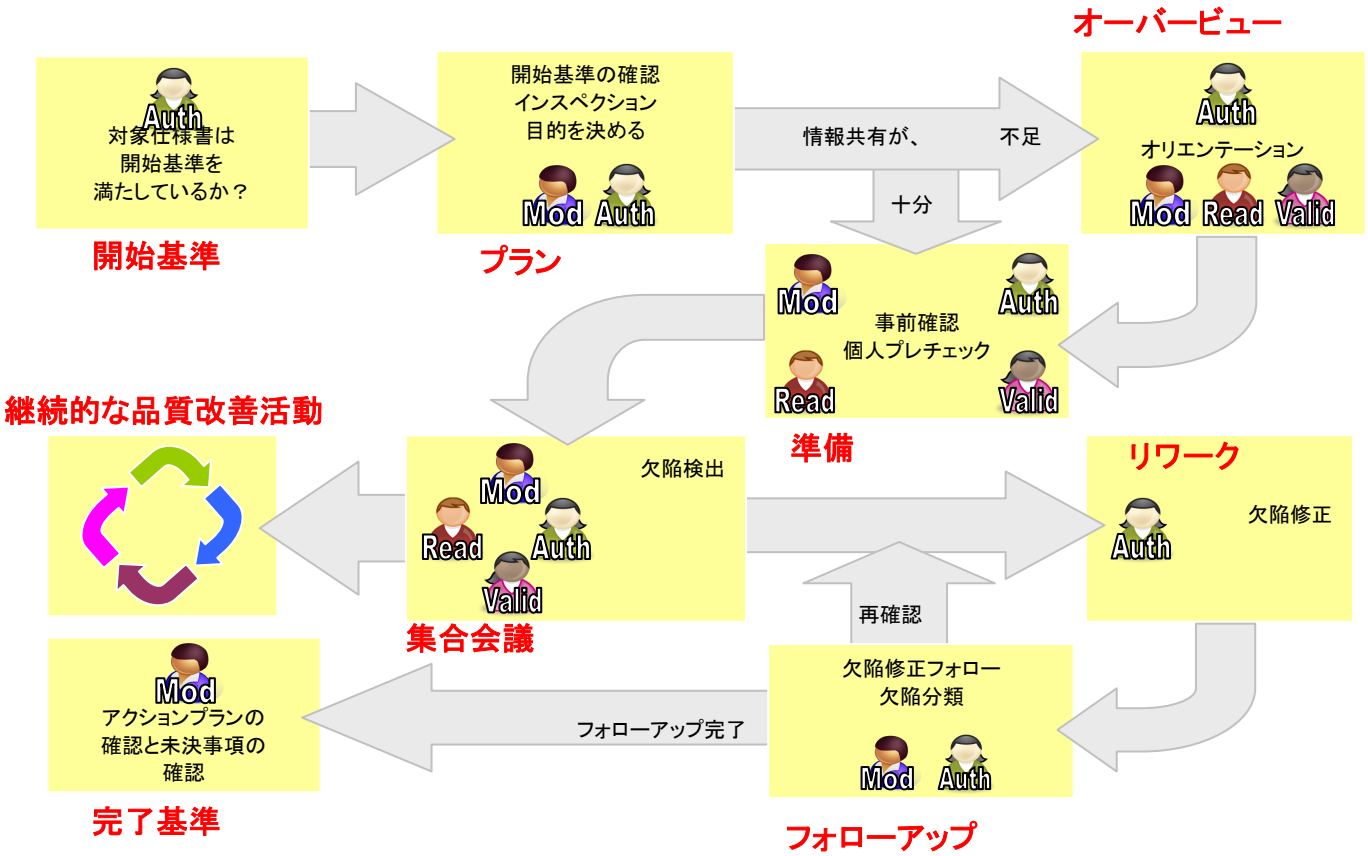
1) 開発者視点担当、ユーザ視点担当
2) 業務ロジック担当、実装方式担当、標準化担当
3) テスト担当、保守性担当、機能性担当、操作性担当

第3章　レビューの演習

　これまで学んできた「レビューの基礎知識」をもとに実際に演習を行って、レビューすることの意味と価値について考えてみよう！今回は「フォーマルインスペクション」を例にレビューの流れを確認していく。

3. 1. フォーマルインスペクションのプロセス

　一般的なフォーマルインスペクションのプロセスは以下のようになっている。実施コストはかかるが、最も欠陥検出・欠陥予防効果の高い手法ということで広く知られている。



3. 2. フォーマルインスペクションの計画

まず、レビュー会議を迷走させないために、レビュー実施前にレビュー計画を立案する。レビュー計画では、レビューの目的と観点を明確にして、「どんな観点でレビューを行うか」を確定しておく。レビューの目的と観点を明確にすることで、参加者個人の経験や、偶然の思いつきに依存する部分が無くなり、毎回均質なレビュー会議が開催できるようになる。

（1）レビューの目的

　レビューの目的は、「致命的欠陥（Operation Defects）」を検出することである。以下に致命的欠陥の例を幾つか挙げる。

・設計上のシステム構造分割誤り
・処理方式誤り
・異常処理の誤り
・インターフェース設計の誤りなど

（2）レビューの観点

　レビューの観点は、どんなバグを致命的欠陥と捉えて検出したいかという様々な「視点」のことを定義することである。以下に観点の例を幾つか挙げる。

・システム利用者観点
例：このシステムを使うことで今抱えている問題を解決できるか？　使い方が分かり易いか？

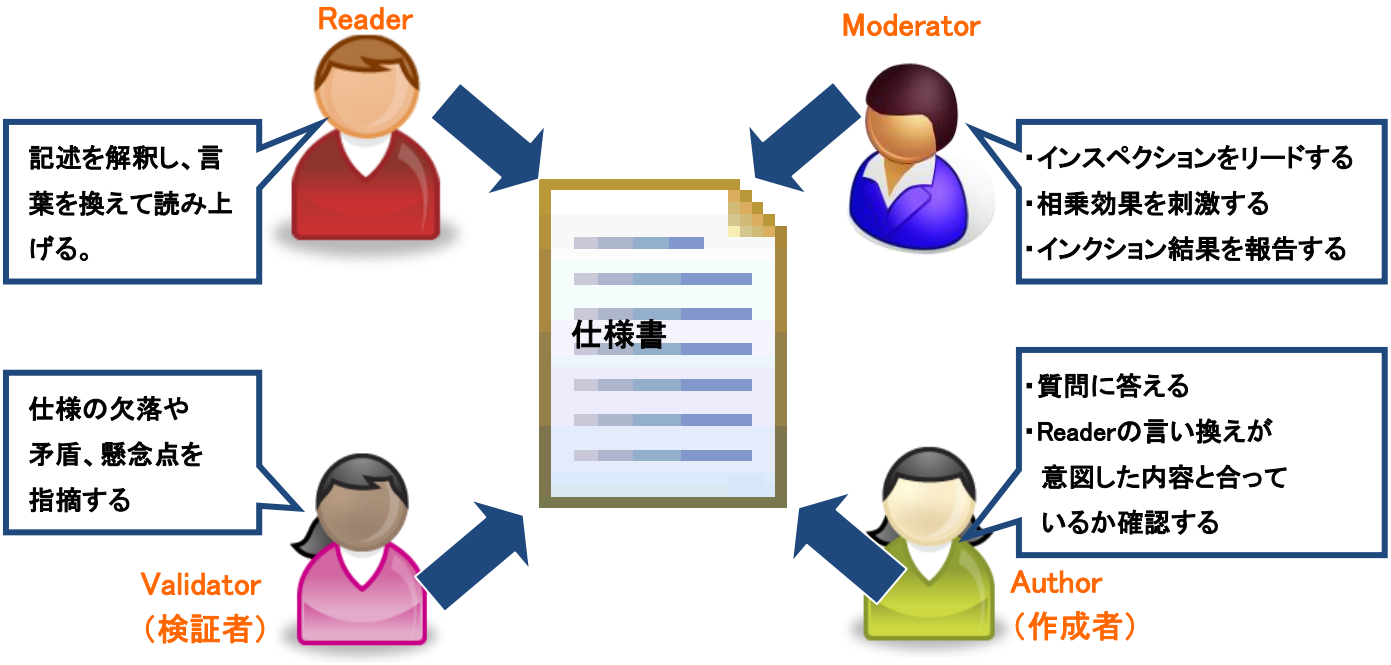
・設計者観点
例：用意された資料でシステムを設計・開発できるか？　仕様の抜け漏れや不整合がないか？

・テスト担当者観点
例：想定外の操作や入力への対策が考慮されているか？　仕様に問題点はないか？

・ドキュメント品質チェック観点など
例：必要な資料が揃っているか？　本行程で必要な作業が全て完了しているか？

3. 3. フォーマルインスペクション実施時の役割

フォーマルインスペクションは以下のように4人一組になってそれぞれの役割を決めて実施する。

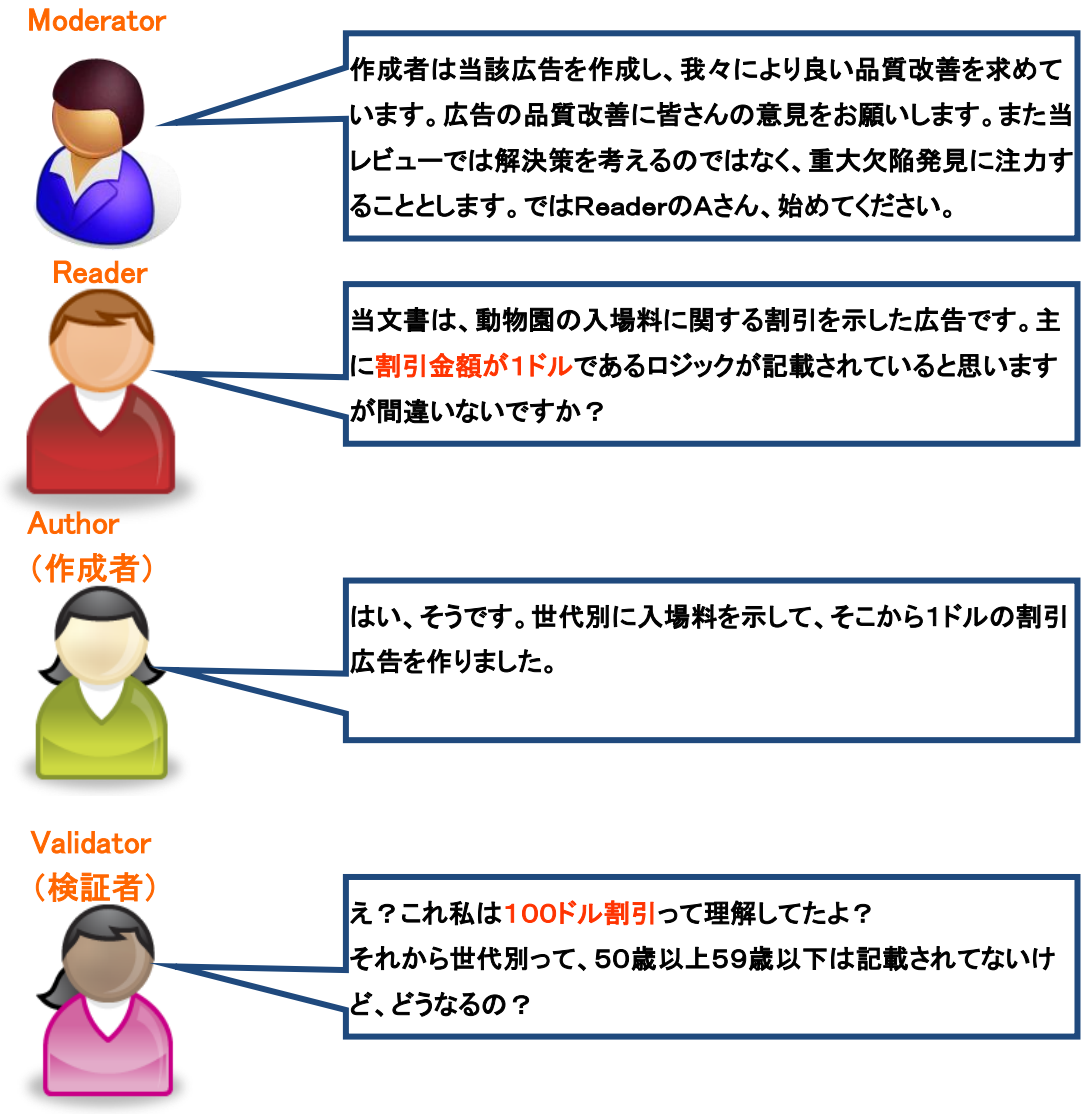


3. 4. フォーマルインスペクションの演習

それでは実際にやって見よう！以下のドキュメントに紛れ込んでいる欠陥を2つ、役割を決めて探してみよう！

3. 5. フォーマルインスペクションのサンプル

フォーマルインスペクションでは、レビュー対象物に記述されている内容をReaderが解釈して言い換えを行い、Validatorが欠陥を指摘する。



第4章　プロジェクトへ導入する際のヒント

レビューをプロジェクトのどの段階で行うか、またどのようなレビューを行うかでレビューの効果は大きく変わってくる。レビューの目的を明らかにし、プロジェクトの各段階でどのようなレビューを行っていくのかを計画していくことは重要である。本章では、プロジェクトの各段階に適したレビューの技法について紹介する。また計画したレビューを実施する際の工夫や、レビューによるコスト削減効果についても紹介する。

4.1.レビュー戦略あってのレビュー計画

計画と言うとタイムスケジュールや進捗管理など、あまり良いイメージを持たない方も多いのではないだろうか。誤解を恐れず言えば、計画は重要ではない。重要なのはどのように品質を確保していくかの戦略を立てることである。開発手法やテストなどと組み合わせてレビューの戦略を決定し、その戦略から計画を作成及び随時更新していくことが出来れば、計画は単なるガントチャートの塊ではなく品質を一つ一つ確保していくための強力な武器となる。

4.2 レビュー計画の作成例

レビュー計画作成時の考慮点について、サンプルを紹介する。

4.2.1 レビュー計画の作成例（タイミング）

レビューの有用性は多くが認めるところだと思うが、実際はやる時間がなくて行っていなかったり、レビューを行なったものの、その成果を反映させるとスケジュールが大幅に遅れることが分かり、結局見送られたなど、有効に活用できていない事案が多い。レビューを効果的に活用するには、プロジェクトの初期段階からレビュー計画を盛り込むことが重要となる。

4.2.2.レビュー計画の作成例（規模）

レビューの規模の面から考えると、大規模なレビューを数回実施するよりも、小規模なレビューを周期的に実施した方が欠陥潜伏期間を最小に出来る。

4.2.3.レビュー計画の作成例（開発プロセスに合わせたレビュー技法の選択）

開発プロセス	レビュー技法	レビュー内容
仕様書作成序盤	アジャイルインスペクション	書いた部分の体裁や記述方法をすり合わせる。 観点例：誤った記述が複数箇所へ展開される前に早期発見する。書けたところから確認する。
仕様書作成中盤	ウォークスルー	自信のないところを局所的に担当者同士ですりあわせる。 観点例：インターフェイス定義など不安部位の確認。短時間で該当部分を確認する。
仕様書作成終盤	インスペクション	全体を通して問題がないかを確認する。 観点例：全体のスループットなど整合性を確認

		する。レビュー対象が揃ってから一貫性を確認する。

4.3.レビューを実施する際の注意点

レビュー実施の際は、以下の点に留意するとよい。

・開催の前にレビュー対象、観点、目標を周知する。
・レビュー開始時に目的と方針を確認し、共有する。
・短時間でもよいので、レビューを定期的（最低週1回）に行うことで、習慣化する。
・レビューアに対して具体的な観点を設定する。
・レビューアの得意分野に沿って観点を各々設定する。
　→指摘の重複を避けると共に、バランスのよいレビューを行うことができる。
・レビュー本来の目的に沿わない議論をレビュー中にしない。

4.5.レビューによるコスト削減効果

欠陥の除去コストは、レビューで見つけた場合を1とすると、テストで見つけた場合は10、ユーザー受け入れテストで見つけた場合は 100 というように、手戻りが大きくなるため下流で見つけるほど除去コストが増大する。

例えばテストで 90 件、受け入れテストで 10 件、計100件の欠陥を見つけていたプロジェクトの欠陥除去コストと、この欠陥の半分の 50 件をレビューで見つけるようにした場合の欠陥除去コストは次のように計算できる。

レビュー無しの場合	検出個数	欠陥除去コスト	トータルコスト
レビュー	0	1	0
テスト	90	10	900
ユーザー受入テスト	10	100	1000
合計	100	—	1900

レビュー有りの場合	検出個数	欠陥除去コスト	トータルコスト
レビュー	50	1	50
テスト	45	10	450
ユーザー受入テスト	5	100	500
合計	100	—	1000

このように、欠陥の半分をレビューで見つけるだけでも、トータルの欠陥除去コストを47%削減できることが期待できる。

第5章　レビュー文化の醸成

4 章まででレビューに関する基礎的な知識と演習および、現場への導入のヒントを示した。5 章ではレビューを継続的に実施して行く上でのヒントを紹介していく

5.1.継続の重要性

レビューを継続することで得られる効果としては、ドキュメントなどレビュー対象の品質向上、チームメンバーのレビューに関するスキルの向上、レビュー実施方法の改善、欠陥検出から欠陥予防へのレビュー目的の変化などが挙げられる。ただしこれらは一朝一夕で得られる効果ではなく、如何にレビューを継続して実施できる環境を作り出せるかが重要になってくる。

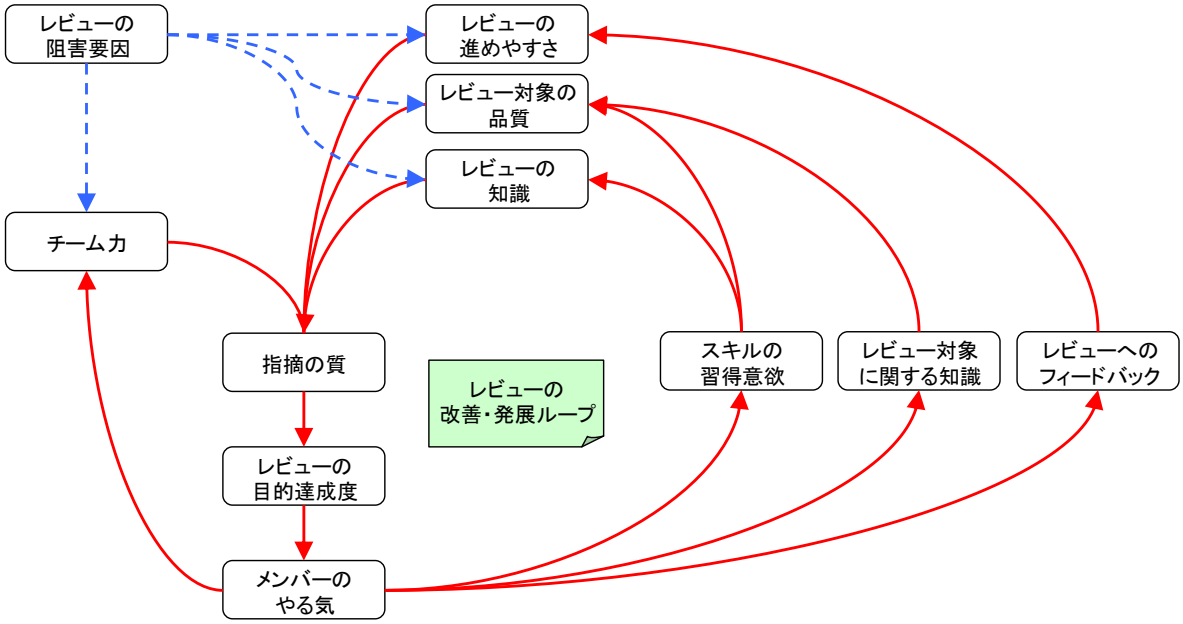


図5－1 レビューの継続ループ例

図5－1はレビューの継続ループ（サイクル）例である。レビュー時の指摘の質が上がることでレビューの目的が達成されやすくなる。その結果メンバーのやる気が向上し、レビューに関連するスキルに対する学習意欲の向上やレビューの進め方に対するフィードバックが増大し、レビューの活動の継続と改善が期待できる。

このループを阻害する要因としては、チーム環境の悪さ、間違ったレビューの進め方、メンバー同士のコミュニケーション不足や信頼不足、モチベーションの低下などが挙げられる。

5.2.レビューのアンチパターン例

以下にレビューに関するアンチパターンを紹介する。レビューに関して問題を感じた場合の参考にして欲しい。

問題	改善案
攻撃的な指摘になる	・相手を尊重する ・開発者のモチベーションを下げずに指摘を反映してもらうことまでが、自分の仕事と考える ・指摘は事実を伝えるのみにする
指摘が反映されない	・時間的、技術的な問題の場合はレビュー実施のタイミングを早期に設定する ・感情的な問題の場合は相手のプライドを傷つけないよう指摘の伝え方を工夫する
発言者が偏る	・一度に発言できるのは一人1件まで、順番に発言する、相手の発言を否定しない等のルール設定や、発言しやすい雰囲気を作る ・参加者にレビュー時の観点や範囲を割り振る
参加者が集まらない	・参加者に直接会ってレビュー実施をアナウンスする ・直接会って事前の状況や出欠を確認する
チェックリストの項目が膨大になる	・1 回のレビューでチェックする項目数を制限する ・チェック項目を変えて複数回レビューする
過去の指摘内容や検出した欠陥情報が、次回以降のレビューで活かされない	・検出した欠陥と、検出に必要な観点をセットで記録する ・欠陥の情報を、利用しやすい形に抽象化する
レビューが長時間化する	・レビューの実施回数を増やし、一回当たりの実施時間を短くする ・レビュー時の観点を絞り、集中してレビューする ・昼食前（午前中）など、確実に中断を入れられる時間帯にレビューを行う ・進行係を決め、欠陥の検出と指摘以外に話がそれないよう監視してもらう
レビューの目的が不明である	・レビュー後にどうなっていたいかを明確にする ・品質計画や開発状況、関係者が感じている懸念事項などから目的を導出する
レビュー時の観点が不明である	・検出すべき欠陥を明確にする ・既存の欠陥情報から観点を設定する
軽微な指摘に時間が割かれる	・誤字脱字等はリストにして担当者に渡す ・重要な指摘から始める
手戻りが大きい	・プロジェクトの早期からレビューを実施する ・重要な仕様からレビューを実施する ・重要人物（有識者など）にレビューへ参加してもらう

レビュー対象のボリュームが多すぎる	・変更された仕様や複雑な仕様など、問題がありそうな範囲からレビューを実施する ・レビューの優先順位を明確にする ・定期的にレビューを実施し、負荷が集中しないようにする
レビュー対象の品質が不足している	・レビュー対象の作成と平行して記述内容をレビューし、レビュー対象の品質を確保する
レビューアの知識が不足している	・上級のレビューアとペアとなってレビューを行い、レビューアに知識を伝承する

5.3.レビュー活動のチェック表

【レビュー文化の醸成に重要な8つのチェック項目】

1，チームワーク	2，レビューの改善	3，計画立案
4，レビューの楽しさ	【レビュー文化の醸成】	5，効果測定
6，モチベーション	7，資産活用	8，実施・習慣化

組織においてレビュー活動を改善していく場合、組織の現状と次のステップへの目標が明確だと改善活動も行いやすくなる。以下にチェック表を用意したので、現状確認と目標設定に役立てて欲しい。

【使用例】

・組織の現状について、何をチェックすれば分からないときの参考にする
・目標を立てる際の参考にする
・チェック表を自分の組織に合うようカスタマイズする
・個々のチェック項目について、各自の理解を話し合ってみる

【チェック項目の見方】

次ページよりチェック表を紹介する。表中では比較的達成が容易な項目を左下に並べているので、始めは左下のチェック項目を参考にレビュー活動を改善していくことを推奨する。ただし順番にチェック項目を達成していく必要はないので、あくまで参考として欲しい。

【チームワーク】

□チームビルディング ◇チーム演習や飲みにケーションなど、チームの結束を高める工夫を持っている	□チームプレイ ◇チーム目標に向けて協調している	□ファントム ◇個人では成し難い、チームならではの成果が出ている
□役割分担 ◇各人が自分の役割を把握し、行動できる	**チームワーク**	□Respect & Influence ◇メンバー間に信頼関係が成立し、お互い良い影響を受ける
□自己紹介 ◇チームメンバーについて、仕事以外で趣味や好みが何か知っている	□ヒューマンコミュニケーション ◇日常会話がある	□テクニカルコミュニケーション ◇技術的な議論が出来る

【レビューの改善】

□ソフトウェア工学の知識習得 ◇プロジェクトに必要な知識を学んでいる	□原因分析 ◇問題点が何かを分析できる	□改善の継続 ◇定期的に振り返りと改善が実施されている
□レビューの知識習得 ◇各作業の意味や目的を理解している	**レビューの改善**	□着実な一歩 ◇小さくても着実に改善点を設定できる
□振り返りの実施 ◇振り返りの機会を設定している	□現状把握 ◇現在の状況を客観的に把握できる	□良い点を見つける ◇レビューが上手くいっている理由を把握できる

【計画立案】

□品質目標とレビュー ◇品質目標達成に適したレビューの内容となっている	□テストとの組合せ ◇レビュー（静的な確認）とテスト（動的な確認）のバランスが取れている	□混入期間の最短化 ◇成果物への欠陥の混入期間が最短となるようレビューが実施されている
□品質目標の明確化 ◇品質目標（確保したい品質特性）が明らかになっている	**計画立案**	□開発段階とレビュー ◇プロジェクトの開発段階に応じて実施内容を調整している
□品質計画への導入 ◇品質計画中に、レビューの計画が含まれている	□工数確保 ◇計画時にレビューの工数が確保されている	□完成度とレビュー ◇成果物作成の序盤と終盤など、成果物の完成度に応じて実施内容を調整している

【レビューの楽しさ】

□ほめる ◇相手の良い点を見つけて伝えることが出来る	□感謝 ◇感謝の意志を伝えられる	□次もやりたい ◇次回もレビューをやりたいと思える環境が出来ている
□自発的な行動の尊重 ◇相手の自発的な行動や意見を否定しない	**レビューの楽しさ**	□フラットな関係 ◇上下関係を持込まない、または持込むメンバーは同席させない
□多様性の尊重 ◇楽しさに対してそれぞれ異なる価値観を持っていることを理解している	□試行錯誤の容認 ◇試行錯誤はより良いレビューの実現に必要であると理解している	□誰でもレビュー可能 ◇参加者にあった役割や観点が用意されている

【効果測定】

□レビューの上達効果 ◇レビュー継続による指摘内容の向上を実感できる	□狙い撃ち ◇レビューの実施タイミングや実施方法、リスク分析など、最も効果を出せそうなレビューを実施できる	□欠陥予防効果 ◇指摘による欠陥の検出から欠陥の予防へと、レビューの活動が高度化している
□間接的メトリクス ◇現場の雰囲気が以前より明るくなっている	**効果測定**	□レビューのコスト削減 ◇手作業の自動化など、より少ない負荷やコストでレビューを実施できる
□欠陥検出効果 ◇レビューで欠陥を検出できている	□検出した欠陥 ◇レビューで重大な欠陥を検出できている	□コスト換算 ◇検出した欠陥から、節約したコストを見積もることが出来る

【モチベーション】

□短期目標の設定 ◇短期的に達成したい目標を持っている	□長期目標の設定 ◇長期的に達成したい目標を持っている	□内発的動機付け ◇外部からの賞罰等に関係なく、モチベーションを自分自身で維持管理できる
□成功体験 ◇レビューで期待した効果が得られてる	**モチベーション**	□ゲーミフィケーション ◇良い働きを褒めやすい仕組みがある
□自分の好み ◇自分の好みを把握してレビューに活かしている	□得意分野の開発 ◇ドメイン知識など、得意なレビュー分野を持っている	□チームワーク ◇チームの一員として協調して作業している

【資産活用】

□記録の整理 ◇記録された情報を利用しやすい形式に整理している	□活用の手段 ◇欠陥別、ステークホルダ別など、利用方法に合わせて情報を活用できる	□資産の整備 ◇資産が陳腐化しないよう、常に整備されている
□資産の活用 ◇記録した資産を実際に活用している	**資産活用**	□社外の知見 ◇他社の取組みなど、社外の知見を利用できるよう情報収集している
□情報の記録 ◇指摘内容や欠陥情報などを記録している	□ベテランの知見 ◇ベテランの知見が誰でも利用できるよう分析整理されている	□社内の知見 ◇横展開など、社内の知見を利用できるよう整理されている

【実施・習慣化】

□誰でもレビュー ◇参加者に適した役割や観点を用意できる	□定期開催 ◇レビューの日など、定期的にレビューを開催するための工夫をしている	□日常化 ◇レビューをするのが当たり前で、しないと不安になる
□レビューの見直し ◇レビュー終了後の振り返りなど、定期的にレビューを見直している	**実施・習慣化**	□ノウハウ ◇レビュー実施のノウハウを複数持っている
□レビューの実施 ◇レビューを実施している	□心構え ◇レビューの重要性を理解している	□標語 ◇レビューのコツを簡潔に表現できる

別冊　レビュー標語録

レビューの実施や改善などにすぐに役立つノウハウ、心構え、標語を紹介する。

標語によっては抽象的な内容でレビュー中級者以上向けのものもあるが、その時々の自分に合った標語を読んでみて欲しい。

(1)ノウハウ

経験豊富で専門知識を持ったレビュー上級者でなくても、有効な指摘をすることは可能である。専門知識保有者でなければ検出不可能な欠陥の存在は否定できないが、「～の場合」というキーワードがあった場合の「～でない場合」の記述漏れや、「A 又は B で無い場合」など二通りに解釈できる表現が無いかなどは、形式的ではあるが、曖昧性や多義性を含むため後工程での検出は難しく、重大な欠陥となる場合が少なくない。これらは、形式的なためコツさえ掴めば誰でも検出可能である。レビュー時の観点を変えたり、やり方を工夫したりすることで、レビュー初心者でも効率的にこのような重大な欠陥を検出することが可能になる。また、逆にレビュー上級者が多数参加している場合でも、レビューのノウハウを知らないと欠陥検出とは無関係な技術論争会になったり、責任を攻め合う場になったり、得るものが少ない結果となってしまうこともある。「ノウハウ」では、有効指摘導出のための観点や、効率的・効果的なレビュー実施のための工夫などを紹介する。

欠陥検出	
ノウハウ-1	**曖昧な表現の言葉を探す(*1)** 言葉の順番を変えたり、表現を変えたりすると違った意味にとれることがある。 例えば、「絶対に巨人に勝ってほしい」は、読み手が巨人ファンか阪神ファンかで意味が変わってくる。 例えば、「向こうから子供が好きなおばさんがやって来た」は、子供がおばさんを好きという意味にも取れるし、おばさんが子供を好きという意味にも取れる。 (*1:SONY 永田敦のアジャイルインスペクションより)
ノウハウ-2	**論理的な曖昧さに注目する(*2)** 言い換えをして、作成者に確認することで、意図している内容に齟齬がないことを確認する。 例えば、「Lサイズで、かつ男性用でない場合は半額です。」を「SかMサイズの女性用は半額です。」と違う言葉で言い換えて、意図した内容になっているかを確認する。この場合、「でない」がLサイズと男性用の両方にかかるのか、それとも男性用だけにかかるのかで意味が変わってくる。 (*2:IBM 細川宣啓のフォーマルインスペクションより)
ノウハウ-3	**レビューでみつけるべき欠陥** レビューでは、見逃した場合に「修正工数が大きい」欠陥を指摘することが望ましい (参考文献[10]より)
ノウハウ-4	**間接的なメトリクスを見る** 「ファイル更新時間」「句読点の数」「開発者の机上に残るペットボトルの数」など一見欠

	陥には関係なさそうなメトリクスでも欠陥が含まれていそうな位置や種類を推測することができる。 例えば、「ファイル更新時間」が深夜の場合、思考が衰えている眠たい時間帯に設計書が作成されたため単純なミスを犯しやすい、上司や有識者が帰宅してしまっており確認が取れず自分で勝手に判断し、上司のレビューを受けていないため、仕様認識の齟齬による誤解釈や考慮漏れなどの欠陥が混入しやすい。 （参考文献[3]より）
ノウハウ-5	**テスト設計をする（*3）** テストケースが作れるか、このテストケースを流して大丈夫か考えながらレビューする。 （*3：IBM 細川宣啓氏より）
ノウハウ-6	**レビュー観点（*4）** ・曖昧ではないか？ ・明確か？ ・矛盾はないか？ ・テスト可能か？ ・設計要素がないか？（要求仕様において） （*4：SONY 永田敦のアジャイルインスペクションより）
ノウハウ-7	**自分の言葉で言い換える** 作成者の文章を自分の言葉（理解）で言い換えてみることで、作成者の認識とのズレが見つかる。
ノウハウ-8	**曖昧な表現の言葉を探す（*5）** 例えば、「など」は、仕様が確定していない可能性あり。「～に同じ」は、本当に同じなのか、どこまで同じなのか。句読点が多い文章は、仕様が複雑または未整理などの可能性がある。 （*5：IBM 細川宣啓氏より）
ノウハウ-9	**欠陥は最後にある（*6）** 最後に作成した設計書は、仕様がなかなか決まらなかったところであり、仕様確定に至っていない可能性もある。 （*6：IBM 細川宣啓氏より）
ノウハウ-10	**利用する場面を想像する（*7）** そのシステムを誰が使うのか、利用する場面を想像してみる。 （*7：IBM 細川宣啓氏より）
ノウハウ-11	**関係する人を徹底的に挙げる（*8）** そのシステムに直接的に関係する人、間接的に関係する人を徹底的に挙げてみる。 （*8：IBM 細川宣啓氏より）
ノウハウ-12	**過去を調べる** 過去の遺産から未来を予測する。同じ失敗は繰り返さない。 過去の類似プロジェクトの欠陥情報や障害情報などを確認し、同種の欠陥はレビューで見逃さない。
ノウハウ-13	**素人になってみる**

	暗黙知や常識を一旦脇に置いて、記述の内容や目的に問題ないか考えてみる。
ノウハウ-14	**長い文章は要注意** 仕様書の文書中に読点(「、」)が何度も出てきて長い文章を記述している場合は、欠陥が含まれていることが少なくない。 (参考文献[9]より)
ノウハウ-15	**ペットボトルや飲料缶の数を見る(*9)** なぜか昔からトラブルプロジェクトでは、開発者の机が空き缶とペットボトルの山になっている場合が多い。 (*9:IBM 細川宣啓氏より)
ノウハウ-16	**コピー&ペースト率を見る** 安易に作成した仕様書にはその貼り付け個所の前後に欠陥が内在しやすい。 (参考文献[9]より)
ノウハウ-17	**プロジェクト全体の目的を満たすこと** 要件定義書をレビューする際には、以下の質問を自問自答してみるとよい。 ・これは要件か？ ・これはプロジェクトの目的達成のために必須か？ ・これが実現しなかったらどうなる？ (参考文献[9]より)
ノウハウ-18	**実現性を確認する** その設計書で、テストケースが導出できるかを考える。 (参考文献[9]より)
ノウハウ-19	**全体像から見る** ソースコードは細かい内容を見る前に次のようなポイントから確認する。 1. 総行数は何行か？ 2. 構造化されているか？ 3. 1 人で書いているか？複数人で修正しているか？ 4. 難しいロジックか？難しい入出力か？ (参考文献[9]より)
ノウハウ-20	**コードのレビュー時にはコードだけを検査しない** コードだけを検査しても、設計書にある「日本語のあいまい性」に起因する欠陥を検出することはできない。そのため、複数人でコードと設計書を付き合わせる。複数人で見ると、設計書の記載内容が複数の意味に解釈できてしまうことに気づきやすい。 (参考文献[9]より)
ノウハウ-21	**テスト計画書は開発関係者全員でレビューする** テストフェーズ開始の時点で全員でテストの方針、技法、標準プロセスの共有を通じて均質化を図らなければ、テストの妥当性・十分性を議論するまで至らない。 (参考文献[9]より)
ノウハウ-22	**時系列変化を追いかける** 最初の仕様書と最後の仕様書が同じ粒度で記載されているかを見る。 (参考文献[9]より)

ノウハウ-23	**直感を大切にする** 「何か」が引っかかったら即確認する。

効果測定	
ノウハウ-24	**欠陥の除去コストは潜伏期間に対して等比級数的に増加する** 検出した欠陥をコスト換算として、欠陥の潜伏期間を使うとよい。ただし、等比例級的に増加する。例えば、設計書に混入した欠陥をレビューで取り除いた場合の手戻りコストに対して、テストで検出した場合は 10 倍、ユーザ受入テストで検出した場合は 100 倍という報告もある。
ノウハウ-25	**欠陥1件あたりの節約コストを示せば効果が見える** 欠陥を1件見逃した場合に、手戻りコストがどれだけ発生するか計算してみせる。

実施方法	
ノウハウ-26	**最初から全部やらないこと、出来る事から始める** 何事もはじめから完璧にやろうとすると労力の割に効果が出ない。準備だけで終わってしまうこともある。最初はすぐ簡単に出来ることから始めるのが良い。
ノウハウ-27	**15 分から始めてみる(*10)** 2時間のレビュー時間を確保するのは難しい。忙しいという理由で休止にされるかもしれない。15分程度であれば無理なく始められるし、単位時間当たりの効果も大きい。 (*10:SONY 永田敦氏より)
ノウハウ-28	**指摘と対応方針の検討は分ける(*11)** レビュー時間は無限ではない。欠陥検出が目的であれば、開発担当者だけで出来る対応方針の検討は後ですべきである。まずは欠陥検出に注力し、指摘漏れがないようにする。 (*11:IBM 細川宣啓氏より)
ノウハウ-29	**1 回の重たいレビューより、軽く複数回実施する(*12)** 一回の負荷を軽くし、複数回実施することで、欠陥検出率よりも「欠陥がすり抜ける率」を低減でき、こまめに検査を行う習慣をメンバー全体に身につけさせる。 (*12:IBM 細川宣啓氏より)

指摘の仕方	
ノウハウ-30	**重要な欠陥から指摘する(*13)** 最も重要と思う指摘を初めに挙げることで、開発者に聞く耳を持たせる。 (*13:静岡大学 森崎修司氏より)
ノウハウ-31	**誤字脱字はレビューの後で(*14)** 誤字脱字はレビューの場では指摘せず、一覧にして渡すだけで良い。 (*14:静岡大学 森崎修司氏より)
ノウハウ-32	**人を責めない(*15)** 人ではなく、成果物に対する指摘を行う。

	(*15:IBM 細川宣啓氏より)
ノウハウ-33	**欠陥を指摘する際に伝える3つのこと(*16)** 欠陥を指摘する際は、どこが欠陥なのか、その欠陥を放置するとどんな影響があるのか、どのように修正するのが良いかの3点を伝える。 (*16:IBM 細川宣啓氏より)
ノウハウ-34	**相手の立場を考えて意見を述べる** 相手の立場を確認してから意見を述べることで、意図が伝わりやすくなる(決めつけで話すと誤解が広がる)
ノウハウ-35	**レビューの場で使ってはならない言葉(*17)** レビュー崩壊の呪文 ・「と思う」「と思われる」 推測や憶測ではなく、事実に基づいて議論しなければ、意味の無い議論となる。 「この機能は他でも使われると思う」と根拠なく言うことで混乱を招くだけなら言うべきではない。 ・「一般的に～」 プロジェクトの特性や状況を考慮して発言すべきである。「一般的に必要な機能だから」、「この機能は、一般的には共通機能にして拡張性を持たせるべき」などと一般論を述べても仕方がない。対象としているプロジェクトやシステム固有の事情を考慮すべきである。 ・「～はおかしい」 何を指摘しているのか正確に伝えなければならない。 どこにどう書いてあるかの事実、その記述の何が悪いのかの理由、そのまま放置しておいた場合どうなるかの影響、この3つを伝えるべきである。 「～はおかしい」、「ここ何か変」、「意味不明」では、伝わらない。 ・「誰が書いたの？」 レビューは欠陥を検出する場である。作成者を責める場ではない。 (*17:IBM 細川宣啓氏より)

（2）心構え

レビューの場では、本来欠陥検出に注力すべきであるにも関わらず、欠陥を混入させたのは誰か、何故混入させてしまったのかを問い正す場になってしまったり、逆に作成者は欠陥を指摘されるのを嫌い自信のあるものだけをレビューの場に持参したり、指摘を受けても素直には受け入れられず言い訳をしたりと、レビューアとレビューイが対立してしまい、建設的なレビューが行われない場合がある。レビューの効果を最大限に得るためには、レビュー参加者の間で良い関係が構築できていることが必要不可欠である。「心構え」では、レビュー参加者間で良い関係を築き、意義あるレビューにするための心構えを学習する。

レビューアの心構え	
心構え-1	**決して人を責めるような言葉を発しない。人ではなく成果物に焦点を当てる(*18)** レビューは吊るしあげの場ではない。成果物の品質を高めるための活動である。人を責めても品質は向上しない。レビューへの抵抗感が高まり品質活動に悪影響を及ぼす。 (*18:IBM 細川宣啓氏より)
心構え-2	**他のレビューアの意見を尊重し、決して否定的な意見を言わないこと(*19)** レビューは攻めあう場でもなく、誰が正しいかを決める場でもない。互いの意見に耳を傾け品質向上に向けて共に協力しあう、尊敬しあう関係を築かなければレビュー品質の向上はない。 (*19:IBM 細川宣啓氏より)
心構え-3	**作成者は限られた時間の中で、成果物を作成していることを忘れてはならない(*20)** 作成する時間が無限にあるのなら高品質のものを作成できるかもしれないが、作成者は限られた時間の中で、成果物を作成しているということをレビューアは分かっていなければならない。 (*20:IBM 細川宣啓氏より)
心構え-4	**必ずしも最高の品質を求めることがゴールではない、プロジェクトやシステムの特性に応じた品質というものがあるということを忘れてはならない(*21)** 人命を扱うシステムと会議室予約システムでは求められる品質が違う。最高の品質を求めることがゴールではない。品質だけでなく、納期・コストも含めた最適が求められていることをレビューアは知っておかなければならない。 (*21:IBM 細川宣啓氏より)
心構え-5	**自分の権威を守るために指摘するのではない(*22)** 自分の権威を守るために指摘するのではなく、検出すべき欠陥があれば指摘をする。必要がなければ全く指摘しないという選択もすべきである。 (*22:IBM 細川宣啓氏より)
心構え-6	**つまらない欠陥と勝手に決め付けて発言しないのはいけない。小さな欠陥に重大な問題が隠れていることもある** 他のレビューアの指摘に比べてあまり重要そうでないから、単なる言葉の間違いだから指摘しなくてもよいと、自分勝手に決め付けるべきではない。直観的におかしいと思ったところには重大な欠陥が潜んでいるものである。ちょっとした言葉のミスが後に大きな誤解となり重大な欠陥が混入する可能性もある。

心構え-7	**技術に環境を合わせるのではなく、環境に合わせて技術を選択するべし** 使える技術に合わせて環境を整えるのではない。それでは進化はない。与えられた環境に対して適切な技術を選択していくべきである。知らない技術であれば習得すればよい。小さい殻に収まっていては成長はない。
心構え-8	**先入観は視野を狭める** 知識や経験を増やすことは大事であるが、知っているが故、それが先入観となり視野が狭くなってしまうことがある。先入観を捨てて過去の経験が全て同じように当てはまると決め付けずに見ることが重大な問題を見つけるために必要なことである。
心構え-9	**「やらされ」ないこと** やらされ感を持ってレビューすると、目的意識が薄く指摘も表面的になりがちである。
心構え-10	**バグを入れたい人はいない** 欠陥を意図的に混入させる人はいない。欠陥を検出しても混入させた人を非難すべきではない。
心構え-11	**常に次回が最高のレビュー** レビューを実施する毎にレビューの質を向上させるべく改善を図るべきである。次回のレビューをもっと効率的・効果的に実施するためにはどうすれば良いかと常に考え、改善していく。
心構え-12	**レビューは、レビューイからレビュー参加者への知識移転の場でもある** レビューの目的は、欠陥を検出することであるが、副次的な効果として知識移転がある。レビューイがレビューアに対して説明することで知識の移転が期待できる。 （参考文献[8]より）

レビューイの心構え	
心構え-13	**レビューは欠陥を見つけ成果物を改善するものである** レビューの目的は欠陥を検出し成果物の品質を高めるための活動であり、作成者の能力を測ったり、作成者を責めたりする場ではない。
心構え-14	**レビューアの指摘を真摯に受け止めること** レビューアは品質を高めようという思いで指摘を行っているのであり、気になるから指摘するのである。指摘は真摯に受け止め改善できる点は素直に改善すべきである。レビューアの認識不足が原因で挙がった指摘ならレビューアを安心させてあげるための説明は行ってあげるべきである。
心構え-15	**自信のあるものより自信の無いものを見てもらう方が得られるものは大きい** 特に品質を高めたいと思う成果物（重要なものや自信のないもの）をレビューで見てもらう方が得られる利益は大きい。レビューに出すために計画以上に成果物の品質を高める行為はレビュー実施に対する負荷を増やすが効果は減らす行為である。
心構え-16	**成果物の最終責任は自分である。レビューアが欠陥を見つけてくれることに頼ってはならない** レビューアが欠陥を見つけてくれるからと自己チェックを疎かにしてはならないし、レビューアが全ての欠陥を見つけてくれるものと勘違いしてはならない。低品質の成果物から

	は重大な欠陥を検出することは難しい、軽微な欠陥が取り除かれ読みやすい成果物であれば重大な欠陥検出に注力できる。また同じ指摘をわざわざ全ての箇所にしてくれるわけではないので、レビューアから指摘された点については他の箇所にも同様の欠陥がないか作成者自身が確認すべきである。

レビューア・レビューイ共通の心構え	
心構え-17	Respect & Influence(*23) お互いに尊敬し影響を及ぼすことをスローガンとして掲げ、お互いに切磋琢磨することでお互いに向上していく気持ちが大事である。 (*23:IBM 細川宣啓氏より)
心構え-18	**レビューを楽しむべし** レビュー対象に興味を持ち、品質向上活動というすばらしい活動を楽しむべきである。

(3)標語

「心構え」で紹介したように、レビューでは実施の目的意識と集団協働作業としての会議参画意識が重要である。また、レビューには、「ノウハウ」で紹介したように、有効指摘導出のための技術観点や、効率的・効果的なレビュー実施のための工夫がある。これらを標語という形で表現することで、親しみやすく覚えやすくする。レビュー開始時にこの標語を掲げて参加者の意識を高めたり、標語を用いて言い難い事を間接的に伝えたり、レビューの場を盛り上げたりするという使い方も可能である。「標語」では、レビューでの心構えやノウハウ・コツを標語にしたものを紹介する。

技術観点	
標語-1	**他人の欠陥は見つけられても、自分の欠陥は見つけられない** 作成者は内容を理解しているため説明不足であることに気がつかない。 作成者は自分が正しいと思っている。他の方法もあることに気がつかない。 作成者は自分が書いた文章を他の人が違う意味で解釈するかもしれないことに気がつかない。 作成者は自分の作品に自信を持っているし思い入れが強い。 これらを克服するためには、他の人に自分の書いた文章をその人の言葉で説明してもらい、それを聞くのがてっとり早い方法である。
標語-2	**部分でなく本質を見ろ(*24)** 例えば、「看板を見て誤りを見つける」という場合、看板に何が書いてあるのか文字を見るのではなく、その看板の材質であったり、その看板が置かれている場所など、本質的なところでおかしなところがないかをまず見るべきである。 (*24:IBM 細川宣啓氏より)
標語-3	**開発者は「木を見て森を見ず」、レビューアに必要なのは「全体を見渡す目」** 専業／分化の進むシステム開発の形態では、「木を見て森を見ず」な開発に陥りやすい。 (参考文献[11]より)
標語-4	**レビューは 1 ページ目から始めてはならない(*25)** まず全体像を捉えよ。段階的にレビュー範囲を狭めよ。 (*25:IBM-QI チーム標語より)
標語-5	**仕様同士の衝突／矛盾に着目せよ(*26)** 仕様同士に矛盾／衝突がある場合、複数人数あるいは何世代かに渡っての改変失敗として捉えよ。 (*26:IBM-QI チーム標語より)
標語-6	**仮説を立てろ。しかし仮説をまず否定せよ。(*27)** 欠陥の位置、種類、欠陥混入背景、原因推定、様々な「仮説」を持ちながらレビューを行う。しかし、仮説は時としてレビューアの目を曇らせる。 (*27:IBM-QI チーム標語より)

心構え	
標語-7	**欠陥を憎んで、人を憎まず** レビューの場では、欠陥検出に注力し、決して人を責めてはならない。 作成者に対して、「なんで、こんな設計をしたんだ」などと責めてはいけない。
標語-8	**手を合わせ、心むなしゅうして、レビューすべし(*28)** レビューアは、レビューを始める前に、手を合わせ、心むなしゅうしなければならない。 レビューアは、一度見直せばすぐ分かるような欠陥が含まれていたり、前回指摘したことが反映されていなかったとしても、決して怒ってはならない。感情的にならずに冷静にレビューするために、レビュー対象物に過剰な期待をしないで望むという心構えが必要である。という有難い教え。 (*28:静岡大学 森崎修司氏より)
標語-9	**レビュー対象を愛せるか、レビューの深さは愛の深さ(*29)** レビューが表面的なもので終わってしまうのかどうかは、レビューアがレビュー対象物に対してどれだけ愛情を持てるかどうかで決まる。作成者がレビューアの指摘を素直に受け入れられるかどうかも、その愛情の深さに比例する。 レビューアがレビュー対象物を愛することが出来ない場合は、否定的な意見ばかりが多くなるし、対象物を深く見ようとしない。その結果、表面的な指摘が多くなるし、指摘の伝え方も否定的な言い方になるため、作成者の心に届かない。 レビューアが、レビュー対象を愛そうとすること、良く知ろうとすることが重要である。 (*29:SONY 永田敦氏より)
標語-10	**エゴレスプログラミング** エゴレス(自我に捕らわれない)プログラミングでは、作成者は一歩下がり、成果物のどこに改善の余地があるのかを他の人々に指摘してもらう。エゴレスプログラミングを実践している人たちは、成果物が他人にとってわかりやすいものであるべきだということを理解している。人間はミスを犯すものであり、時に外からの見方が役に立つことも認めている。エゴレスなレビューアは、同僚に対する思いやりと感受性を持っている。 (参考文献[1]より)
標語-11	**作者は、欠陥検出をレビューに過度に期待してはならない。最終責任は自分にある。** プログラマの中にテスト担当者がエラーを見つけてくれるのを期待する向きがあるのと同様に、誰か他人が自分の誤りを見つけてくれることに頼ってしまって仕事が甘くなる人がいる。成果物の最終責任は作成者にある。 (参考文献[1]より)
標語-12	**レビューは完成前に行う作業** 作成者は、自分以外の目が作成物を見ることを許す前に完璧な出来栄えにしたいという誘惑に駆られる。しかし、全て完成してからレビューするという考え方は間違っている。完璧なものを仕上げるのはレビューを通してから。レビュー期間、指摘の修正期間、再レビュー期間もスケジュールに必ず盛り込むことを忘れずに。 (参考文献[1]より)
標語-13	**対立的でない言葉遣いで指摘を伝えるためには、「あなた」ではなく「私」** レビューアは問題を提起する際に使用する言葉を、成果物について観察したことに焦点

	を当てて慎重に選ぶべきである。「これらの変数がどこで初期化されたのか、私はわからなかったのだけれど」と言えば、建設的な回答を引き出せそうである。それを「あなた、これらの変数を初期化していませんよ」では、作成者が気を悪くするかもしれない。非難めいた「あなた」から、より対立的でない「私」へ、言葉遣いを少し変えることで、レビューアは批判的なフィードバックでも効果的に伝えることができる。 （参考文献[1]より）
標語-14	**自ら定期検診に行く人は、行かない人より総じて健康である** 本人の意識が（プロジェクトの）健康にとって大事である。
標語-15	**バグとは過ちと見過ごしの積み重ねであり、常識の死角に隠れている** バグは自然発生しない、原因は自分の内にある。
標語-16	**過ちは等しく訪れる、自分だけ逃れることは出来ない（でも真っ先に相手を疑ってないか？）** 人は自分で自分自身の犯した過ちに気づきにくく自分は完璧だと思いがちであり、つい他人を疑ってしまう。しかし人は誰でも過ちを犯すものである。
標語-17	**まず自分を疑え、相手はその後である** 人は問題を見つけると自分以外に原因があるとつい思ってしまう。問題を見つけたらまず自分に原因が無いかを疑うべきである。相手を疑うのは簡単だが、本当に自分には原因が無かったのかを確認してからにしよう。
標語-18	**バグを憎んで人を憎まず** バグの混入原因と予防に集中する。バグを混入したい人はいない。
標語-19	**トカゲが怒る** ついカッとなった時、怒っているのは自分の中の別の生き物であると考える。本来の自分は問題に対して感情でなく理性で解決を図る紳士である。
標語-20	**理解の齟齬はバグの素** レビューで理解の共有化を図る。
標語-21	**多くの悲劇は使えない技術を選択したことではなく、選択した技術の使い方を誤ることで生じる** どんないい道具でも使い方を誤ったり過大な期待を抱いている間は効果は出ない。また問題は道具でなく使用者にある場合も十分あり得る。
標語-22	**“レビュー”とは、責め合うものではなく、品質を高め合うもの** レビューの目的と指摘内容がはっきりしていれば、人を責める取り組みではないと理解できるし、個人的な感情論に陥ることも防げる。 （参考文献[1]より）
標語-23	**レビューは背水の陣であることを念頭におく事（*30）** 最も高効果と高コストである目視レビューは一般に有効であると信じられているが、レビューで見落とした欠陥は致命傷になる確率が高い。 （*30：IBM-QI チーム標語より）
標語-24	**全ての欠陥には重みと偏りがあることを念頭におくこと（*31）** （*31：IBM-QI チーム標語より）

標語-25	**全ての欠陥を除去できない(*32)** (*32:IBM-QI チーム標語より)
標語-26	**全ての欠陥は人工物／人間の産物である(*33)** (*33:IBM-QI チーム標語より)
標語-27	**信頼すれども信用せず(*34)** レビューを一緒に行う仲間(含:開発者／設計者)を信頼せよ。しかし(品質に関して)信用するな。 (*34:IBM-QI チーム標語より)
標語-28	**時間のないプロジェクト程、レビューを重視せよ(*35)** (*35:IBM-QI チーム標語より)
標語-29	**成果物は嘘を付かない。人が嘘を付く(*36)** (*36:IBM-QI チーム標語より)
標語-30	**レビューの場に、政治／権力は不要である。常にフラットであれ。(*37)** (*37:IBM-QI チーム標語より)
標語-31	**眠い時はレビューをするな。体力維持も睡眠もプロの仕事の一部(*38)** (*38:IBM-QI チーム標語より)
標語-32	**テストはリアクティブ、レビューはプロアクティブ** テストは作り込んだ欠陥を後から取り除くというのに対して、レビューは作りこむ前に取り除くということである。

文化	
標語-33	**スイカが割れた(*39)** レビューの場で、意見が発散してしまった状態を表す言葉。 レビューでは欠陥検出のみに注力すべきだが、レビューアが本来議論すべきところから外れた話を始めたら、「スイカが割れた」と言って、議論を本来の欠陥検出に戻してあげる。 (*39:WACATE というワークショップのグループで、レビューアが集まった夏合宿でスイカ割りをした際にスイカが激しく飛び散ったことを議論の発散に例えたエピソードから来た標語。このように参加者に共通の話題を元にして比喩を用いると場が盛り上がる。)
標語-34	**1日1回、歯磨きするようにレビューしよう(*40)** 歯磨きは誰しも毎日行うもの。レビューも歯磨きと同じように、毎日のように行うのが理想的である。レビューはまとめて実施するのではなく、毎日 15 分だけでも少しずつレビューを実施する。その日に作成した設計書はその日に欠陥を取り除く。1日の終わりに(もしくは1日の初めに)「今日は歯磨きした？」とレビュー実施の有無を確認する。レビューしないと何か気持ち悪いと思えるようになったらレビュー文化が根付いた証拠である。 (*40:SONY 永田敦氏より)
標語-35	**プロジェクトを遅らせるのは、欠陥である** レビューなどはコストのかけ過ぎであり、プロジェクトを遅らせると思い込んでいる人がいる。実際には、レビューがプロジェクトを遅らせたりはしない。欠陥が遅らせるのである。

	欠陥を修正するための手戻りコストは総開発工数の 40～60%を占めることがわかったという報告もある。レビューへ投資することで、コストが高い後工程での手戻りの総量を減少させることができる。 （参考文献[1]より）
標語-36	**急がばレビュー** レビューには時間がかかる。しかし、レビューを賢明に適用すれば、いくつかのテスト段階がバイパスできて、実際には製品開発スケジュールを短縮し得る。もしレビューに割く時間が無いと言っていると、テスト担当者や顧客が見つけた欠陥を修正するためにさらに多くの時間が必要になる。 （参考文献[1]より）
標語-37	**レビューで得られる教訓は、テストで得られる教訓より、遥かに大きく、開発プロセス強化に役立てるべきである** レビューによって蓄積した経験は、開発プロセスの強化のために役立てなくてはいけない。そうすればチームメンバーの作業の誤りは少なくなり、より多くの時間を節約することになる。 （参考文献[1]より）
標語-38	**レビュー結果の個人評価への利用は、「レビュー文化の殺し屋」** ・レビュー結果のせいで罰せられないように、成果物をレビューに提出しなくなる ・レビューアは作成者の敵と思われたくないから指摘しなくなる ・本当に欠陥なのか討論することになる ・欠陥が表面化しないように、あまり発見しないようにすることが暗黙の目標になってしまう ・1回のレビューで発見される欠陥数を減らすため、非効率と知りながら成果物を小分けにしてレビューするようになる （参考文献[1]より）
標語-39	**バグの利息は破産に繋がる** バグを放っておくと大変な目に遭う。
標語-40	**予防は治療に勝り、意識は予防を促す** バグは入れないのが一番。
標語-41	**食後は歯磨き、作ったらレビュー** 作った端からレビューすれば怖いもの無し。
標語-42	**呼出せファントム！(*41)** チームワークならではの効用。1人では検出できないが、複数人で意見を出し合っているうちに自分だけでは気づかないであろう問題点が誘発されるように見つけることができることがある。 (*41：IBM 細川宣啓のフォーマルインスペクションより)
標語-43	**一人だと苦行でも集まると楽になる** 一人で黙々と指摘をあげていくとイライラしやすいが、集って作業すると、不明点を解説してくれたり「やっぱりそうだよねー」と共感が得られるので辛くなくなる。
標語-44	**何回「なるほど！」出ましたか？**

	グッドな指摘や思いもよらなかった観点など、レビュー参加者の活躍に対して素直に賞賛しよう。
標語-45	**出来は何点？** レビューに対する振り返り。次回に高得点を取るには何が必要か考える。
標語-46	**継続は上達の素** 初めは思うように指摘が出なくても、ベテランの指摘内容や必要な観点が分ってくれば少しづつレビューの腕前が上達していく。
標語-47	**レビューで得られる知識は開発の比にならない** 1ヶ月開発するよりも、1週間レビューする方が、遥かに多くの欠陥パターンを知ることができるし、より合理的なプログラミングを体感することができる。
標語-48	**バグは記録せよ。バグは企業財産である。(*42)** 全てのレビューアは、「赤ペン」「ポストイット」「バグ手帳」を携行せよ。 (*42:IBM-QI チーム標語より)
標語-49	**指摘をするな。成長させよ。(*43)** 赤ペンの指摘書き込みは、読み手が成長するためのアドバイスであれ。決して「上から目線の指摘」であってはならない。 (*43:IBM-QI チーム標語より)
標語-50	**集中する方法より、集中が途切れる瞬間を自覚せよ。(*44)** 集中力を持続する技法を追求するより、集中が途切れる瞬間を自覚した方が、長時間のレビューが可能になる。適度の Break とそのタイミングが大切。 (*44:IBM-QI チーム標語より)
標語-51	**今日の 1 欠陥除去より、明日の 10 欠陥予防。最重要課題は同種欠陥の再発予防。(*45)** (*45:IBM-QI チーム標語より)
標語-52	**それが欠陥に見えるのはレビューしているあなたの視点(*46)** (*46:IBM-QI チーム標語より)
標語-53	**欠陥を教えるな。測り方／見つけ方を教えよ。(*47)** (*47:IBM-QI チーム標語より)
標語-54	**レビューでの知識移転／教育は、目的ではない。副産物である(*48)** (*48:IBM-QI チーム標語より)

その他	
標語-55	**書いてある欠陥を見つけられたら初級、書いてない欠陥を見つけられたら上級(*49)** 書いてある欠陥を見つけることよりも、書いてない欠陥を見つけることの方がよっぽど難しい。レビューアは見えないものを見る力を養う必要がある。想像力、妄想力こそがレビューアに必要な資質である。 (*49:IBM 細川宣啓氏より)
標語-56	**指摘は重要なものから伝え、誤字脱字はリストで渡せ(*50)** 指摘を伝えるときは、プロジェクトにとって最も重要なもの、インパクトが大きいものから

	伝え、誤字脱字のようなものはレビューの場で伝えるのではなく、後で一覧にしたリストを渡すのが良い。軽微な欠陥から伝えた場合、作成者に不快感を与えるだけである。「どうでもいい指摘をしやがって」など思われないように、作成者が最も凍りつくような最も有難い指摘から伝えることによって、レビューアの威厳も保たれるし、聞く耳を持ってくれるようになる。 (*50:IBM 細川宣啓氏より)
標語-57	**レビューの目的は欠陥を検出することではない。欠陥を入れないようにすることである。(*51)** レビューは欠陥を検出すれば良いと思っている人がいるかもしれないが、それはレベル1である。欠陥が無い品質の高い成果物を作成できるようにするのが本来目指すところである。 レベル1:欠陥を検出する レベル2:指摘した欠陥を修正させる レベル3:指摘したもの以外の欠陥も修正させる レベル4:成果物の品質を上げる レベル5:作成者の品質を上げる (*51:IBM 細川宣啓氏より)
標語-58	**立ち上がり重視の2・8レビュー(*52)** 作成する設計書全体の2割が出来たところでレビューする。これによって、ある一定の品質が確保できる。そして、まだ修正する期間が十分残されているであろう全体の8割が完成したところでレビューする。これによって全体の整合なども見据えた確認を行う。 (*52:IBM 細川宣啓氏より)
標語-59	**1枚でもレビュー(*53)** 全て完成してからレビューを行うのではなく、1ページでも出来ていればレビューは出来る。こまめにレビューすることで手戻りも最小限に留めることが出来るし、品質を向上させていくこともできる。 (*53:SONY 永田敦のアジャイルインスペクションより)
標語-60	**困ったときはダッキー君(*54)** ダッキー君に相談することで、自分の考えが整理され、息抜きになる。ストレスや不安は溜め込まないこと。 (*54:IBM 細川宣啓氏より)
標語-61	**レビューで安心、出来ました?** レビューで自分が不安に感じている箇所を解決することが出来れば、品質向上に繋がっていく。 (参考文献[2]より)
標語-62	**事実のみを指摘せよ。感覚/感情/過去の経験は一切指摘するな。(*55)** (*55:IBM-QI チーム標語より)
標語-63	**欠陥は表現出来た時に初めて欠陥として成立する。(*56)** 品質保証部のアヒル部長に、検出した欠陥を簡潔に説明せよ。説明できたら指摘候補として成立。

	(*56:IBM-QI チーム標語より)
標語-64	**「その欠陥は今除去すべきか？」を添えて指摘すること。(*57)** (*57:IBM-QI チーム標語より)
標語-65	**解決策の提示はレビューの場以外で(*58)** 検出にのみ注力するのがレビュー指摘率を上げるコツ。 (*58:IBM-QI チーム標語より)

参考文献

[1]「ピアレビュー」 日経BPソフトプレス社 Karl E.Wiegers (著)

[2]「ソフトウェア品質不安に対する心理的側面に着目した、レビュー計画作成技法の提案」 2010年度SQiP研究会 第3分科会 細川 宣啓、永田 敦、森崎 修司、山本 浩之、牧野 将治、小原 美帆、奥山 剛、小田部 健

[3]「間接的メトリクスを用いて欠陥予測を行うレビュー方法の提案」 2010年度SQiP研究会 第3分科会 細川 宣啓、永田 敦、森崎 修司、諏訪 博紀、中谷 一樹、田邊 哉好、末次 努、森崎 一邦

[4]「修正確認テスト規模の低減を目的としたコードレビュー手法」 情報処理学会論文誌Vol. 50 No. 12 3074-3083 (Dec. 2009) 田村晃一、亀井靖高、上野秀剛、森崎修司、松本健一

[5]Radice, R.A, N.K. Roth, A.C.O'Hara. Jr. and W.A. Ciarfella,

[6]A Programming process Architecture, IBM Systems Journal, Vol.24, No2 (1985)

[7]細谷、吉岡、森崎「産学連携によるデザインレビュー改善事例」 http://www.jaspic.org/event/2011/SPIJapan/session1B/1B3_ID003.pdf

[8]IEEE 1028 Standard for Software Reviews and Audits - IEEE Std 1028?-2008 (Revision of IEEE Std 1028-1997)

[9]ThinkIT 連載:インスペクションは誰が行うべきか 第 4 回 インスペクタはこんなところをみている by 細川宣啓 http://thinkit.co.jp/article/901/1

[10]ThinkIT 連載:技法とテンプレート！第 4 回 インスペクションで何を指摘するべきか by 森崎修司 http://thinkit.co.jp/article/896/1

[11]ThinkIT 連載:インスペクションは誰が行うべきか 第三者インスペクションとは 本記事のまとめ by 細川宣啓 http://thinkit.co.jp/article/857/1?page=0,2

[12]@IT ソフトウェアレビューのワークショップ開催 by 細川宣啓,森崎修司 http://www.atmarkit.co.jp/news/200907/03/re.html