

| レビューの技法                    |                          |                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                               |
|----------------------------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| No.                        | レビュー名                    | 手法                                                                                                                                                                                                            | 留意点                                                                                                                                                                                                                                                                                                                                           |
| <b>3.8.1 レビュー方法</b>        |                          |                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                               |
| 3.8.1.1                    | ピアレビュー                   | 同僚を意味するピアによるレビュー全般の呼称である<br><目的>作成者とは別の視点から、レビューを気軽かつ頻繁に実施できるようにする                                                                                                                                            | ・技術者間でノウハウを共有できる<br>・技術者間でのコミュニケーションを促進できる<br>個人評価の観点からレビューに含めてはならない。管理者や上司が参加する場合は、「レビュー結果を個人評価に利用しない」と明確に宣言する<br>とよい                                                                                                                                                                                                                        |
| 3.8.1.2                    | インスペクション                 | 標準仕様から外れた例外、文書化された標準からの逸脱や文書化された要求仕様から見た障害などを発見するレビュー方法の一つであり、最も公式なレビューである<br><目的>障害発生防止と障害発生後の後戻り工数の削減により、高い品質目標の達成とプロジェクトリーダーでのコスト低減を実現する。                                                                  | 障害が早期に発見され、後戻り工数が少く障害を修正できる<br>効果的なインスペクションを実施するためには、守るべきルールや要求仕様を公式に文書化しておく必要がある                                                                                                                                                                                                                                                             |
| 3.8.1.3                    | チームレビュー                  | チームにより実施されるレビュー。効果的な人数は8人から15人程度とされる。                                                                                                                                                                         | ・最も公式なレビュー方法であるインスペクションに比べ効果は落ちるが、低コストで効果の高い障害検出が期待できる<br>・レビューチーム内での知識共有、プロジェクト構成員の教育ができる                                                                                                                                                                                                                                                    |
| 3.8.1.4                    | ペアプログラミング                | 二人が一つのマシンを共有して協同でプログラミングを行う方法であり、レビュー方法の一つとしても位置付けられる<br><目的>プログラムの作成時に他者の視点を常に取り入れることによる、コードの理解性向上、コードレベルでの詳細設計の理解性向上、コードに修正込まれる障害の早期除去                                                                      | ベテランと若手を結ぶことで熟練プログラマーの技術をベアの技術者に継承できる。開発チームの士気を向上させるなど、教育効果やビューマンファクターに関する効果が得られる場合が多い                                                                                                                                                                                                                                                        |
| 3.8.1.5                    | ピアデスクチェック                | インスペクションなどの公式なレビューよりも低いコストでレビューを実施する<br><目的>インスペクションなどの公式なレビューよりも低コストでレビューを実施する。                                                                                                                              | コストの低減に加え、グループレビューに比べると作成者の心理的負担が軽減できる。熟練レビューの持つ成果物チェックのポイントを作成者が共有できるなどの効果が挙げられる                                                                                                                                                                                                                                                             |
| 3.8.1.6                    | パスアラウンド                  | レビュー対象となる開発成果物を複数のレビューアへ配布、または回覧を行うことで、障害の指摘を行う形態のレビューである<br><目的>時間制約の問題により対面によるレビューが困難な場合に、レビューア全員にレビュー対象物を配布し同時並行的にレビューしてもらうことで、レビューコメントを併せ。                                                                | 対面形式のレビュー開催が困難な場合でも、レビューができる。また、レビューアに対する時間的制約が低減され、回覧内容を複数の人員で行レビューするため、1人のレビュー遅れがスケジュール遅延の原因にならない<br>・レビューが終了するまでに時間がかかるため、進行状況のフォローが必要である<br>・レビュー間で意見の対立が生じた場合は、別途対面レビューを設定するなどの対応を採るとよい                                                                                                                                                  |
| 3.8.1.7                    | サウンドロブレビュー               | ・異なる視点によりレビューを行うことで、レビュー対象物への理解が深まり、より適切な指摘が行える<br>・さまざまな専門技術を持つ複数レビューアの参加により、多角的な指摘が得られる<br><目的>参加者全員それぞれが、レビュー対象物への理解を深めて、より適切なレビューを行う。また、複数のレビューによる複数の視点からのコメントを併せすることで、多角的なレビューを行う。                       | ・異なる視点によりレビューを行うことで、レビュー対象物への理解が深まり、より適切な指摘が行える<br>・さまざまな専門技術を持つ複数レビューアの参加により、多角的な指摘が得られる<br>・参加者全員にコメントする機会が訪れるため、レビューの傍観者が作られるのを防止できる<br>・参加者のレベルにばらつきが大きいと、レビューの進行が遅くなった<br>・適切な障害指摘ができなかったり<br>・全員に司会者となる均等に機会が訪れるようにテーマを配分する必要がある<br>・特に専門性が求められるレビューに関してはインスペクションなど、別の方法の採用を検討する<br>とよい                                                 |
| 3.8.1.8                    | ウォークスルー                  | 仕様書やソースコードなどの成果物に対し、レビューを含め複数人が内容を確認することで実施するレビューである<br><目的>成果物を含んでいる障害の発見が第一であるが、それ以外にも成果物の理解促進、実装の代替手段の検討、問題解決のアイデア出しなどの目的に使用する                                                                             | インスペクション同様、障害が早期に発見できるため、後戻り工数が少く障害を修正できる。インスペクションに比べ少ない工数で実施できる。また、参加者がさまざまな意見を交換することから、対象物の理解が促進され、問題に対する新たなアイデアが発見できる                                                                                                                                                                                                                      |
| 3.8.1.9                    | アドックレビュー                 | 自らの問題を解決するために、近くの人を呼び寄せて意見を借りるという程度の非公式なレビュー<br><目的>レビューミーティングを開催するまでもない問題に対して、近くの人からコメントしてもらうことにより、第三者的な視点から問題の解決を図る                                                                                         | 容易に第三者の視点からの指摘が得られる<br>個人の自らの問題を解決するための効果しか持たないため、チームやプロジェクト全体の問題に関しては、公式度合いの高いレビュー方法を検討する必要がある                                                                                                                                                                                                                                               |
| <b>3.8.2 仕様・コードに基づいた技法</b> |                          |                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                               |
| 3.8.2.1                    | 形式手法に基づくレビュー             | 形式手法の考え方を活用し、仕様上の矛盾を数学的証明によって検出し、一貫性を確保するためのレビュー技法である<br><目的>厳しい安全性が求められるクリティカルシステムにおいて、システムの一貫性、完全性、正当性を確保する。                                                                                                | 従来のテスト技法では完全な障害除去は期待できないが、形式手法は可能である<br>利用範囲は、以下の理由から、明確に仕様で定義可能な領域に限定する<br>とよい<br>・形式手法を用いた仕様の数学的表現は理解が難しい<br>・对人的な要素が大きき場合は、不確定要素が大きく事前の仕様定義が難しい                                                                                                                                                                                            |
| 3.8.2.2                    | インターフェース分析               | システムを構成するモジュールやクラス、またはシステム間のインタフェースの切り口をつなぐ方法をレビューにより確認する技法である<br><目的>インタフェースにはタイミングと情報があり、レビューによりタイミングの誤り、またはデータ形式及び値の誤りを確認する。                                                                               | 情報の静的なインタフェースを分析する場合には、モジュール構成図やクラス図などを用いて正確性を確認するための基準とすることにより、システム全体のインタフェース確認ができる<br>最上位から最下位のモジュールすべてを通じて確認する<br>とよい。大規模なモジュールについては階層呼び出しモジュールについては少なくとも1階層以上の考慮を要する。インタフェース項目には、機能として取り扱う情報の他、制御のための情報も含まれることがあり、これをまとめること確認し、インタフェース項目の文字、数値、制御情報などのデータ形式、及び長さ、位置、オフセットや並び順について、画面や他システムなどの外部インタフェースと、対象システム全体を通過して一致していることを確認する<br>とよい |
| 3.8.2.3                    | 複雑度分析                    | 複雑度のメトリクスに着目し、ソフトウェアの開発、変更、保守をする際にそれらの困難性を評価するレビュー技法である<br><目的>定量化された複雑性を用いることで、ソフトウェア開発工程における作業の難易度を評価し、必要に応じて複雑度が低下するように設計の手戻りを検討する、あるいはレビュー密度やテスト密度を基準より高くするなどの調整を行う。                                      | 作業の困難性を一貫性のある尺度により評価できるため、作業成果物の品質確保のために強化すべき再設計、レビュー、テストなどの作業を特定できる<br>複雑度の評価基準は、絶対的なものがあるわけではなく組織ごとに相対的なものと捉えること。従って、現場での適用の中心が基準となるデータを蓄積していく活動が必要となる                                                                                                                                                                                      |
| 3.8.2.4                    | パストレース                   | 処理中のパスと、条件によるそのパスの組み合わせとしてのロジックを確認するレビュー技法である<br><目的>対象となるモジュールに含まれている実行可能な経路をパスとして細分化し、そのパスに従って、判断分岐や変数の内容の変化を網羅的にトレースすることにより、ロジック妥当性を確認する。                                                                  | 処理の正当性を確認できる。また、外部仕様との対応の確認や、条件に沿った処理の組み立ての妥当性が確認できる<br>パス数が多くなると効率が悪くなる。パス数を削減するために下位モジュールをブラックボックスとしてみる技法を適用する<br>とよい。また、切り出されたロジックの確認方法として、関数表現、画面を色分けするカラフルマップ技法などを使う<br>とよい                                                                                                                                                              |
| 3.8.2.5                    | ラン・スルー                   | 要求仕様から想定される具体的な入力データを元に、レビューが机上で内部仕様やプログラムを追跡するレビュー技法である<br><目的>要求仕様から想定して頻繁に使用されるプログラムのパスに対して、内部仕様やプログラムの正確性を確認する。                                                                                           | 要求仕様から想定した具体的な入力データを使用することにより、リストにおける初期のトラブルを削減できる<br>ラン・スルー自体のレビューの網羅性は高いものの、他のレビュー技法と組み合わせる<br>とよい                                                                                                                                                                                                                                          |
| 3.8.2.6                    | 制御フロー分析                  | プログラムの制御フロー（IF-THEN-ELSEなどの論理条件）に着目し、そのプログラムをレビューする技法である<br><目的>プログラムの制御フローの正確性や効率性、保守性などを確認する。                                                                                                               | 複雑すぎる制御フローはレビューで検証するよりも、再設計を行った方がよい場合もある                                                                                                                                                                                                                                                                                                      |
| 3.8.2.7                    | アルゴリズム分析                 | ソフトウェアで実現するアルゴリズムに着目し、そのアルゴリズムの正確性や効率性などをレビューする技法である<br><目的>アルゴリズムの正確性や効率性を確保する。                                                                                                                              | ・アルゴリズムの決定は、ソフトウェアの開発に比較的大きな影響を与えるため、できるだけ早期にレビューを行った方がよい<br>・論理的に正しく効率的なアルゴリズムであっても、実装やテストが難しい場合には、代替を検討する<br>とよい                                                                                                                                                                                                                            |
| 3.8.2.8                    | モジュール展開                  | 分割されているモジュールを一つの図面に展開し、モジュール分けの妥当性、インタフェースの設定の適切性をレビューする技法である<br><目的>モジュール分けの妥当性、インタフェース設定の適切性を確認する。                                                                                                          | ・冗長処理や重複処理の発見及び改善ができる<br>・共通データの発見及び改善ができる<br>・保守性の向上ができる<br>大規模なプログラムでは、量が多くなりすぎて全体の見通しが悪くなるため、他の技法を用いる<br>とよい                                                                                                                                                                                                                               |
| 3.8.2.9                    | 七つの設計原理(高橋)              | 障害を作り込まないために考慮すべきプログラム構造上の七つのポイントにまとめた技法である<br><目的>障害を作り込まないためのポイントに沿ってレビューを実施する。                                                                                                                             | 障害の作り込みを検出し、防止できる                                                                                                                                                                                                                                                                                                                             |
| 3.8.2.10                   | 静的解析                     | ソフトウェアを実行せずにソフトウェアの特定の側面や性質を解析する技法の総称である<br><目的>ソフトウェアの特定の側面や性質を解析して、品質の現状を確認し、将来の品質改善の必要性を把握する。                                                                                                              | ツールを用いることで、人手によるレビューでは扱うことが困難な大規模かつ複雑なソフトウェアを解析できる<br>・対象ソフトウェアを改善するためのプロセスや体制を確立しておく必要がある<br>・対象ソフトウェアを実行しないため、実行の動作環境では実行されない箇所も解析する。また、実行時の並行などに起因する非決定性振る舞いや、解析できない領域がある                                                                                                                                                                  |
| <b>3.8.3 フォルトに基づいた技法</b>   |                          |                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                               |
| 3.8.3.1                    | ソフトウェアFMEA/FMECA         | ソフトウェアFMEA: システム中のあるアイテムの故障モードに着目し、その原因調査、影響調査を行うことで、システムの信頼性を定量的に解析する技法である<br>・FMECA: FMEAに付加して、故障発生時の確率及び故障による影響の重大さの格付けを考慮する技法である<br><目的>過去のソフトウェア開発において、欠陥パターンと検討の優先順位を踏つことにより、致命的なものから順番に類似障害を検出し、予防できる。 | 将来のソフトウェア開発において、欠陥パターンと検討の優先順位を活用することにより、致命的なものから順番に類似障害を検出し、予防できる                                                                                                                                                                                                                                                                            |
| 3.8.3.2                    | FTA                      | ディンダベリリ予測において基本的な解析技法として使用される<br><目的>ディンダベリリ予測において基本的な解析技法として使用される                                                                                                                                            | FTAではその発生が好ましくない事象について、発生経路、発生原因及び発生確率をフォルトの木を用いて解析できる<br>動作時間や動作順序に依存するシステムでは、FTAの適用が難しい                                                                                                                                                                                                                                                     |
| 3.8.3.3                    | EMEA(エラーモード故障解析)         | <目的>ハードウェア構成要素について、各々の障害やアプリケーションとの関連による影響を分析しあらかじめ対応策を講ずることにより、システムの可用性を向上させる。                                                                                                                               | エラーモードを見つけ、それらをヒューマンエラーの影響、及び致命度を系統的に見つけることができる。また、その原因から、ヒューマンエラーによる障害を低減するための設計上の考慮点を検討する。これにより、使用性や信頼性を向上させる効果が得られる                                                                                                                                                                                                                        |
| 3.8.3.4                    | CFIA(構成要素障害影響分析) [田村]    | システムにおけるハードウェア構成の脆弱性を発見し、障害時に取るべきアクションを明確化するために用いられる技法である                                                                                                                                                     | 確認内容に応じた適切なステークホルダーによる議論により、効果的、効率的な対応策が検討できる                                                                                                                                                                                                                                                                                                 |
| 3.8.3.5                    | PQ(バグ・キューブ)デザインレビュー [日出] | 顧客一時仕様や設計に想定されるべき部分の障害でシステム全体がダウンする、あるいは主機能が使えなくなるようなシステム障害の未然防止のために実施するシステム的なレビューの技法である<br><目的>PQ障害原因の早期抽出と、システムとして障害復旧性、回復性がより込まれていること、作り込まれた回復機能が一般使用に十分な性能を有することを確認する                                     | システムを実際に構築する前に、顧客の信頼性要求や開発側の意図した使い方で実現できていることを、専門家が集まってレビューすることにより、信頼性の高いシステムを効率的、効果的に構築できる<br>公共性の高い大規模システムではPQレビューは必須である                                                                                                                                                                                                                    |