

第3分科会（第2グループ）

検出難易度の高い欠陥を検出するレビュー観点の提案

The proposal of the review point of view which detects the defect in difficult detection

主査	細川 宣啓	日本アイ・ピー・エム（株）
副主査	永田 敦	ソニー（株）
副主査	藤原 雅明	東芝ソリューション（株）
アドバイザー	森崎 修司	（国）静岡大学
研究員	川合 大之	（株）日立ソリューションズ
	奥山 剛	（株）山武
	上野 直樹	三菱電機コントロールソフトウェア（株）
	會見 知史	（株）インテック
	菅野 良太	（株）リンクレア

研究概要

レビューを実施しても、誤字脱字などの軽微な指摘が多く、時間とコストを掛けた割に効果が少ないと感じているケースが多い。本来、レビューで検出したいのは、後工程で発覚した場合の手戻りが大きい欠陥である。

その中でも関係者全員が見落とす様な検出難易度の高い欠陥は、後工程で発覚する事が多く、場合によっては市場へ流出した後に発覚する事となり、手戻りコストは高く付く。

そこで我々は、レビューやテストで取り除くことが出来なかった検出難易度の高い欠陥に注目し、これらをレビューで検出するための観点を検討した。検出困難な欠陥事例は、各研究員が市場に流出してしまった欠陥やサービスイン後に発覚した欠陥を持ち寄り、特徴や傾向の分析を行った。

これらの事例から、要件定義や外部仕様段階で埋め込まれた欠陥を抽出し、まず、欠陥内容について分類を行った。そこから、欠陥内容を抽象化し、それらを要件定義や外部仕様段階で検出するには、レビューでどの様な視点・観点を持つと指摘が可能となるかについて検討した。

本研究では、検出難易度の高い欠陥を検出するための観点を明らかにし、レビューアの経験・スキル依存せずに、欠陥を指摘するためのレビュー方法について考察する。

Abstract

Even if there is much easy indication even if it reviews, and it hangs time and cost, it is thought that there are few effects in many cases. Originally, that wants to detect in a review is a defect in which the correction at the time of being revealed at a back process is large. As for many defects in which the detection difficulty which all the people involved cannot detect is high, being revealed at a back process is. It will be revealed after flowing into a market depending on the case, and correction cost occurs.

We observed the defect in which the detection difficulty which was able to be removed by neither a review nor examination was high. And How to remove these in a review was examined.

Each researcher gathered the defect which has flowed into the market, and the defect revealed behind service in, and the defective example with difficult detection conducted analysis of the feature or a tendency.

The defect embedded in the requirement definition or the external specification stage was extracted from these examples, and the classification about defective contents was performed first.

We abstracted defect contents. And it was considered what kind of viewpoint and viewpoint it should have. In this paper, we clarify a point of view to detect the high defect of the detection degree of difficulty. And we considered the review technique for pointing out a defect, without an experiencing and skill depending.

1 テーマ選定と背景

近年のソフトウェア開発は、短納期・低価格・高信頼性に対して、ユーザーから厳しい要求が求められている。高信頼性を確保する方法は様々に存在するが、機械的に確認できない部分は最終的に人が確認してこそ品質の高いソフトウェアが完成する。レビューはその一手法であり、欠陥の残存期間を短くし、開発期間を短縮する手法として、期待されている。

しかし、レビューを効果的に運用できている組織は多くない。欠陥をレビューやテストで検出できず市場に流出した場合は、手戻りコストが大きくなり、顧客からの信頼を失う原因にもなる。

そこで、今回の研究は、レビューア全員が見落としとして市場に流出した検出難易度の高い欠陥を「重大欠陥」と定義し、過去の欠陥事例から重大欠陥が混入する原因を推定し、検出する観点を明確化・抽象化することにより、レビューで重大欠陥を検出する方法や検出アプローチを考察した。

2 研究のアプローチ

我々は、想定される事象から考えるトップダウン方式と実際に現場で起こった事例から考えるボトムアップ方式のどちらのアプローチを採用するか検討した。研究グループのメンバー特性は、実際に現場で開発業務を行っており、市場に流出してしまった欠陥を目の当たりにした事例を経験していることから、理想のレビュー方法を導く為にボトムアップで欠陥事例の調査を行うことにした。

収集した欠陥事例の中には、要件定義の工程で混入された欠陥や、設計・実装の工程で混入された欠陥など様々であった。我々は、まず重大欠陥を検出するには、より上流工程で検出すべきだと考えた。その理由は、設計・実装の基となる対象が要件定義や外部仕様であり、設計・実装時のレビュー観点としては主に、「記載された内容通りに設計・実装がされているか」に集約される。一方、要件定義や外部仕様の対象は、人の思いや構想であり、人によって認識の差異が発生する可能性は高くなる。認識の差異は文書上の欠陥検出と比較して検出難易度が高く、重大欠陥へと発展しやすいと考えた。

そこで本研究では、重大欠陥を検出するにはより効果的と思われる「要件定義」と「外部仕様」に着目することにした。

3 重大欠陥の事例調査

様々なプロジェクトから重大欠陥の事例を収集し、重大欠陥が混入する原因を推定・考察した。

3.1 欠陥事例の収集

5人の9～22年の開発経験を持つ、異なる組織に所属するエンジニアが、自組織や自身の業務分野における76件の重大欠陥を持ち寄った。欠陥の発見工程は、システムテスト完了後であり、顧客による受け入れテストやサービス開始後に発見された欠陥である。対象システムは、組込みシステムおよび基幹情報システムである。欠陥の混入工程は、要件定義、または外部仕様作成時である。

欠陥の収集は、表1の収集フォーマットを用いて、著者ら各自の所属企業で行った。表1の各項目の意味を下記に示す。

- ・ 欠陥事例：欠陥の内容
- ・ 混入工程：要件定義、または外部仕様
- ・ 原因：なぜ、欠陥が混入したかの原因
- ・ 対策案：どうすれば、レビューで指摘できたかという対策案

表1 欠陥の収集フォーマット

No	欠陥事例	混入工程	原因	対策案

3.2 欠陥事例の検討

収集した重大欠陥に対して、その欠陥の検出はレビューで行うべきか、テストで行うべきかを検討した。検討の結果、全76件の欠陥中、レビューで検出すべき欠陥は22%にあたる17件であった。

この検討の中で、レビューで検出すべき欠陥としたのは、テストで検出することが困難な欠陥である。例えば、要件に不整合があったが、そのまま外部仕様として記述された場合、外部仕様書に基づいて作成されたテストで、その欠陥を検出するのは困難である。また、レビュー、テストのどちらでも検出可能な場合は、よりコストが安くなる検出方法を選択した。例えば、システム間通信メッセージの外部仕様において、特定ビットの持つ意味に関する記述に誤りがないかを確認する場合、テスト環境の構築が容易であると仮定すると、レビューにて人手で整合性を確認するより、テストにてその挙動を確認する方が安価である。

持ち寄った欠陥を検討する際、欠陥情報を共有することに多くの時間を要した。欠陥情報の伝達には、欠陥そのものの情報だけでは不十分であり、「プロジェクト背景」「制約条件」および「前提事項」の説明が必要であるという知見を得た。

3.3 欠陥混入の原因分析

レビューで検出すべき欠陥に対して、なぜ欠陥が混入したのか、真の原因を検討することにより、下記の2つの原因に集約することができた。

1. 認識の齟齬（15件）
 - 当該機能に対して、ユーザーが求めているものと開発者が想定したものに差異があった（6件）
 - 運用環境のインフラや連携する他システム（開発対象スコープ外）の挙動が想定外だった（5件）
 - 顧客の曖昧な要望に対して、前提の確認を行わなかったために、要望を誤って解釈した（4件）
2. 抜け・漏れ（2件）
 - 顧客の要望から要件定義、要件定義から外部仕様に変換する際に、抜け・漏れが発生していた。（2件）

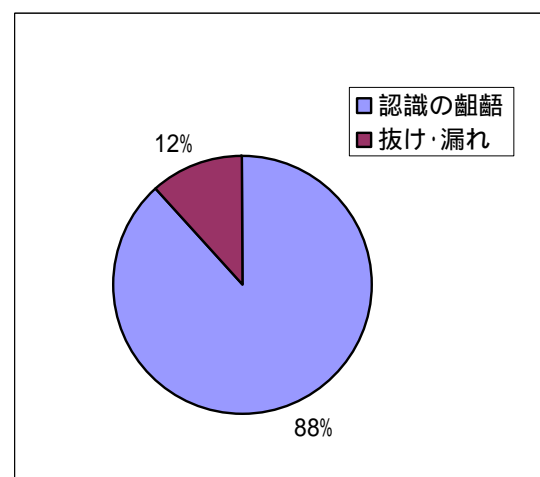


図1 重大欠陥の原因

4 重大欠陥を検出するために必要なレビュー観点

4.1 抽象化したレビュー観点

3章の事例調査の結果より、我々は重大欠陥の混入原因を「認識の齟齬」と「抜けと漏れ」の2つに大別したが、「抜けと漏れ」の問題に関してはソフトウェアトレーサビリティ^[1]など、その解決策が広く知られているため今回の検討から除外し、「認識の齟齬」に着目した。

「認識の齟齬」とは、開発者と顧客との間での認識の齟齬であり、それは主に要件定義書や外部仕様書の記載されていない箇所に対する認識の違いによって発生していると考えた。この書かれていない部分に起因する、認識の齟齬を検出するために必要なレビュー観点を、欠陥事例ごとに検討した。得られたレビュー観点を分類し、抽象化した結果、以下の3つに集約できた。

- ・ ビジネス目的との整合性

そのシステム仕様はビジネス目的と不整合を起こしていないか？

- ・ 暗黙の前提 / 制約

その仕様にシステムの暗黙の前提 / 制約が存在しないか？

- ・ 解決策から生み出される新たな問題の混入

その解決策は新たな問題を孕んでいないか？

次節で、これらの観点について説明する。

4.1.1 ビジネス目的との整合性

我々は顧客のビジネスのある一部分をシステム化する事によって、合理化・省力化を提供している。そのシステムの要件定義書や外部仕様書には、顧客にとって当たり前となっているビジネスドメイン中の常識（暗黙知）は記述されていない場合が多い。また、開発者にとってシステム設計上、当たり前前と考えている事柄も記述されていない場合が多い。これらの書かれていない箇所に対して、顧客はビジネス全体から見た場合に最も合理的で整合性の取れた動作を期待するが、開発者はそのシステムの中での論理的な整合性のみで判断している場合が多く、顧客と開発者との間に認識の齟齬が存在していると考えた。そこで、開発者は仕様が顧客のビジネス全体から見て、その目的と不整合を起こしていないかどうか、という観点が必要である。

4.1.2 暗黙の前提 / 制約

顧客のビジネスの一部分を切り出してシステム化する際に、暗黙の前提や制約などの存在により、その境界線が曖昧な場合がある。例えば、システムのインフラ・基盤やデータ量・パフォーマンス・耐障害性などの暗黙の非機能要件、またシステム間通信において、データの識別番号が連番であるか否かの暗黙の機能要件などがある。そのため、開発者は仕様が、暗黙の前提 / 制約に依存していないか、という観点が必要である。

4.1.3 解決策から生み出される新たな問題の混入

仕様の変更や、自社にとっての新技术採用など、ある問題を解決させるために講じた策が、新たな問題を内包させてしまう場合がある。そのため、その解決策が生み出す新たな問題は混入していないか、という観点が必要である。

4.2 具体的なレビュー観点の例

4.1章で導出した、「ビジネス目的との整合性」「暗黙の前提 / 制約」「解決策から生み出される新たな問題の混入」と言う抽象的な観点から、欠陥の実例を元に、より具体的なレビュー観点を検討した結果、以下の様な観点が挙がった。

これらの観点はアプリケーションドメインに依存しないため、一般的なあらゆるタイプのプロジェクトに適用が可能であり、かつ、観点の導入コストは極めて低いため、現場プロジェクトの計画立案

時（特に品質計画立案時）に有用であるを考える。

表2 具体的なレビュー観点の例

抽象的な観点	より具体的な観点の一例
ビジネス目的との 整合性	<ul style="list-style-type: none">・その機能の意図、存在理由、背景は明確か？・その機能を使用する場면을顧客と合意しているか？・その機能を使用する場面の想定が限定的ではないか？ また、その想定は運用上、守られるのか？・誤操作時(想定外の操作)によるガード、気づけるか？ 誤操作した事に顧客は気づけるか？・誤操作後にリカバリ方法が用意されているか？
暗黙の前提 / 制約	<ul style="list-style-type: none">・その仕様に前提、制約を暗黙的に置いていないか？ その前提・制約が運用上、保証されない可能性はないか？・インフラ・基盤に暗黙の前提、制約が隠されていないか？・他システムI/Fでは、他システム側の問題（未稼働、障害）による 影響を考慮しているか？・他システムI/Fで想定している範囲が明確になっているか？・高負荷、低レスポンス状態での影響は考慮されているか？・ユーザーがその処理で扱う想定データ量を確認したか？ また、そのデータ量を扱うのに適したユーザーインターフェースか？
解決策から生み出 される新たな問題 の混入	<ul style="list-style-type: none">・その解決策を講じることで、周辺機能に影響はないか？ また、パフォーマンス、レスポンスを劣化させないか？・その解決策を講じることで、当初の要件を曲解していないか？ 解決させる為に開発者事由で要望・要件を曲解していないか？・その解決策を講じることで、業務の負荷が増えていないか？

5 予備評価試行

5.1 試行概要

本論文での手法の試行を目的として、極めて小規模な2つの類似した仕様書を用いて、重大欠陥が指摘可能であるか予備試行を実施し、観点の説明前後で重大欠陥の指摘件数が変化するかどうかを確認した。なお、現実には、本論文で想定する重大欠陥は、予備試行の仕様書には含まれていないことが想定されるが、実務者を長時間確保して評価を実施することや、機密漏洩なく実システムと同等の規模の仕様書を試行対象とすることが難しいと判断したため、小規模な仕様書を用いて試行を実施した。

5.2 試行手順

被験者が仕様書1をレビューする。

レビュー観点：軽微な欠陥は無視し、重大な欠陥を検出する。

レビュー時間：20分

4章で示した重大欠陥を検出するために必要な観点を説明する。

被験者が仕様書2をレビューする。

レビュー観点：説明した観点を活用し、軽微な欠陥は無視し、重大な欠陥を検出する。

レビュー時間：20分

アンケートを収集する。（S/W開発経験年数、レビュー経験、感想）

5.3 試行結果

試行の結果を表3に示す。図2に、仕様書1と仕様書2（重大欠陥を検出するために必要な観点を説明した前後）での重大欠陥の指摘件数の変化を示す。なお、詳細な試行結果は、付録1「予備評価試行 結果詳細」を参照のこと。

図2から読み取れるように、初心者および上級者に変化はみられなかったが、中級者の重大欠陥の指摘件数が増加することが分かった。

表3 レビュー経験毎の指摘件数、重大欠陥の平均検出件数

レビュー経験	被験者数(人)	指摘件数(件)		重大欠陥検出件数(件)	
		仕様書1	仕様書2	仕様書1	仕様書2
初心者	5	7.2	7.8	0.8	1.0
中級者	3	5.7	9.3	2.0	4.0
上級者	2	7.0	8.0	3.0	3.0

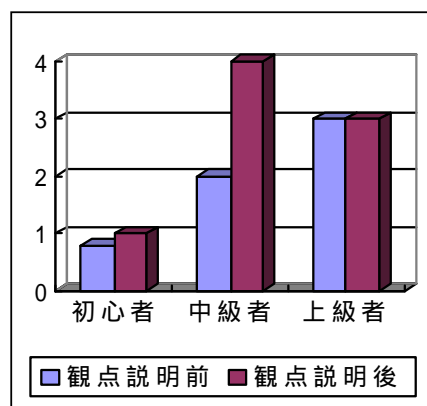


図2 重大欠陥の平均検出件数の変化

5.4 試行結果の考察

レビュー経験の豊富な上級者は、4章で示した観点の説明なしでも重大欠陥を十分検出していた。アンケート結果から既にレビュー観点として持っていることが分かった（アンケート結果は付録1「予備評価試行 結果詳細」を参照のこと）。レビュー経験の少ない初心者は、指摘件数、重大欠陥の検出件数ともに、ほとんど変化はみられなかった。これは、S/W開発経験が少なく、知見として持っている過去の欠陥事例が少ないため、4章で示した抽象的観点から、具体的なレビュー観点を設定することができなかったためであると推測する。一方で、中級者は、指摘件数、重大欠陥検出件数ともに増加しており、効果があると推測する。

以上のことから、4章で示した抽象的観点をを用いることで、レビュー中級者は重大欠陥の指摘件数が増えることが示唆された。

6 重大欠陥を検出するためのレビュー方法

3章の事例調査の結果から、重大欠陥を摘出するための観点を4章で例示した。しかし、この観点を持つことのみで、全てのレビューアが重大欠陥を指摘できるようになるとは言い難く、5章の予備評価試行の結果にもある通り、レビュー初心者には、まだ困難である。

4章で挙げた観点は抽象的であり、欠陥を検出する際に、レビューアのスキル・経験が必要となる。レビューアはレビュー対象に関連する知見をもとに、整合性を確認することや、過去に経験した欠陥事例とパターンマッチングすることで、欠陥を検出する。逆説的に述べれば、経験不足のレビューアであるほど、欠陥を検出することが難しくなる。

そこで、本章では、レビューアの経験・スキルに依存せずに、重大欠陥を指摘するためのレビュー方法について述べる。

6.1 モデレータ^[2]としての品質エンジニアのレビュー参画

重大欠陥を検出するための観点について、検討を進めていく中で、レビュー全体をハンドリングする役割であるモデレータの重要性を再発見した。

我々の開発現場では、モデレータを配置せずに、作成者が進行役としてレビューを実施しているプロジェクトが多かった。これは、当初、モデレータを配置していた場合でも、レビュー対象を見ることが、本来のモデレータとしての役割を忘れ、欠陥の検出に専念してしまうためである。

4章で挙げた「ビジネス目的との整合性」「暗黙の前提/制約」「解決策から生み出される新たな問題の混入」の観点で、レビュー対象の妥当性を確認するには、一步引いた目でレビュー対象を見る必要がある。我々の開発現場では、作成提案、見積、要件定義を経験し、仕様を熟知しているプロジェクトマネージャ/リーダーがレビューアの役割を兼任していることが多かった。このことは一見レビューアとして適切で、品質向上に寄与すると思われるが、様々な外因により 認知バイアスを受け、重大欠陥を検出できない可能性が高まるのである。例えば、レビューアの直前作業、他者の指摘に影響を受け、考慮する範囲が限定されたり、プロジェクト開始から時間が経ち、ビジネス目的を見失ったり、何らかの前提/制約を補足したりして、検出が漏れるといった現象である。

そこで、モデレータを配置し、一步引いた立場で、4章で挙げたレビュー観点を持つこと、及びレビュー対象物の説明者に「言い換え」を促すことで、暗黙の前提/制約が隠されていないか観察することで、検出漏れを防ぐことができると考えた。

別の観点ではモデレータを務める適任者についても検討した。検討の結果、数多くのプロジェクトで客観的に品質を評価してきた品質エンジニアが、上述のモデレータの役割を担うのが最適であると考えた。なぜなら、品質エンジニアは、過去の欠陥事例を数多く蓄積しており、4章で挙げた抽象観点を業種毎に具体化して提示することができるからである。また品質エンジニアは横断的にプロジェクトに参画することが多く、レビュー対象のプロジェクトに専従していないために認知バイアスの影響を受けにくい。よって、プロジェクトマネージャ/リーダーに比べ、ビジネス目的や暗黙の前提/制約、新たな問題の混入について一步引いた目で妥当性を確認できる。品質エンジニアが直接的に欠陥を検出するのではなく、質問することで、レビューアの「発想を転換」し、「気づき」を生む。その結果、レビューアが欠陥を検出できるのである。

以上のことから、品質エンジニアがモデレータとしてレビューに参画することで、レビューアの経験・スキルに依存せずに、重大欠陥を検出するレビューが実施できると考えた。

6.2 網羅的にまとめられた欠陥事例の参照

品質エンジニアによるレビュー参画が難しく、レビューアの経験が不足している場合、過去の欠陥事例とのパターンマッチングによる欠陥検出が難しくなる。この場合、過去の欠陥事例を収集し、具体的な観点として提示することができれば、その代替えとすることができる。

我々の開発現場では、バグトラッキングシステム(BTS)を使用しているプロジェクトは多く、欠陥事例の蓄積は十分できているが、欠陥検出に十分活用されているとは言い難い。全ての開発現場で発生した欠陥事例を収集し、網羅的・体系的に示した「欠陥標本」は存在していない。

「欠陥標本」が存在すると仮定すると、以下の方法で、多くのレビューアが重大欠陥の検出が可能になると考えた。

- ・ レビュー計画で検出したい重大欠陥を定義し、検出するための観点を共有化する

レビュー計画において、4章で挙げた観点を記載し、プロジェクトメンバに展開する。その際、レビューキット^[3]を利用し、観点の共有化だけでなく、レビューで検出したい重大欠陥の定義（レビューの目的）、レビュー時の役割についても共有化しておくことが重要である。

- ・ 「欠陥標本」から過去の欠陥事例を抽出し、レビュー観点チェックリストを作成する

レビューアの経験・スキル不足による欠陥の検出漏れを防ぐために有効なのが、チェックリストを使用したレビュー^[4]である。レビュー対象の特性をもとに、「欠陥標本」から過去の欠陥事例を抽出し、重点を置くべきレビュー観点チェックリストを作成・共有する。

以上のことから、欠陥研究が進み、網羅的な「欠陥標本」の完成が期待される。

7 おわりに

本研究は、市場に流出していた検出難易度の高い重大な欠陥をレビューで検出する観点を提案した。過去の欠陥事例から、「開発者と顧客間での認識の齟齬」が検出難易度の高い欠陥を混入させる大きな原因の1つであることが判明し、更に、欠陥を検出するための観点を検討した結果、「ビジネス目的の整合性」「暗黙の前提/制約」「解決策から生み出される新たな問題の混入」に集約できた。

ただし、全てのレビューアがこれらの観点を活用することは困難である。そこで、これらの観点と過去の欠陥事例を持ち、第三者視点でレビューに参画できる品質エンジニアがモデレータの役割を担うことで重大欠陥の検出が可能になることについて考察した。

また、品質エンジニアの参画が難しい場合や、レビューアの経験不足を補う方法として、過去の欠陥事例を網羅的・体系的に示した「欠陥標本」が有効であると考察した。

今回の研究では、効果の検証は、予備評価に留まってしまったが、引き続き研究員各自の開発プロジェクトにおいて、提案した観点の効果を確認していくと共に、今後は、欠陥分類と検出方法の網羅的な検討を目指す。

参考文献

- [1] IPA SEC, 「高信頼化ソフトウェアのための開発手法ガイドブック」, 2010年3月.
- [2] M.E. Fagan, “Advances in Software Inspection,”
IEEE Transaction on Software Engineering, vol. 12, no. 7, pp. 744-751(1986)
- [3] 細川宣啓、永田敦、藤原雅明、森崎修司、添田建太郎、西村英俊、小田部健、中谷一樹「レビューオリエンテーションキットを用いた育成によるレビュー文化の醸成」, 日本科学技術連盟 ソフトウェア品質研究会, (2012)
- [4] 坂東祐司、森崎修司、松本健一「セキュリティ要件のレビューにおけるチェックリストの表記方法の比較」, 情報処理学会研究報告, Vol.2010-SE-170 No.5(2010)

付録1 予備評価試行 結果詳細

1 指摘件数の詳細

項番	被験者名	レビュー技術	開発経験年数	指摘件数		重大欠陥(*1)指摘件数	
				仕様書 1	仕様書 2	仕様書 1	仕様書 2
1	Aさん	中級者	8年	3	6	3	5
2	Bさん	初心者	3年	7	6	0	0
3	Cさん	初心者	1年	5	5	1	1
4	Dさん	初心者	1年	5	7	1	1
5	Eさん	中級者	4年	4	10	2	4
6	Fさん	初心者	3年	10	11	1	2
7	Gさん	初心者	3年	9	10	1	1
8	Hさん	上級者	20年	8	10	3	3
9	Iさん	中級者	14年	10	12	1	3
10	Jさん	上級者	11年	6	6	3	3

*1 誤脱字、文章表現、文言の曖昧さ、欠陥に直結しない提案 / 質問を除く妥当な指摘

2 アンケート結果

被験者名	感想
A、B、Cさん	・初回のレビューをしているときは、間違え探しの単純作業のような気がした。観点を聞いてから行った2回目のレビューは、知的な楽しい作業だった。
Dさん	・観点が抽象的過ぎて、経験がないために指摘することが難しかった。
Eさん	<ul style="list-style-type: none"> ・観点を念頭において、レビューするのは難しい。手元にリストとして置いておけば、思い出して、考えをめぐらすことができる(ドキュメントを読み込むと、そのシステム化した際のイメージ生成に入ってしまう)。 ・指摘件数が増え、指摘のレベルも上がった印象がある。
Fさん	<ul style="list-style-type: none"> ・具体的な観点が記載されているチェックリストがあった方が属人性も排除できるし、レビューア自身の漏れ・抜けもなくなるので、欲しいところ。 ・Eさんと同じで、その観点を忘れそうになる。
Gさん	・暗黙の前提 / 制約は今まで意識してなかったので、観点が増えて勉強になった。
Hさん	<ul style="list-style-type: none"> ・実験ということで、どこまで指摘すれば良いか迷った。 ・このような観点をチェックリスト化して、仕事で使った方がよい。
Iさん	<ul style="list-style-type: none"> ・普段は、組込みシステムをやっているの、今回のレビュー対象のシステムはイメージしづかった。 ・ビジネス的な観点は、職務の範囲外なので、普段は、レビュー観点には入れていない。
Jさん	・あまり変化なし。初回のレビューでも要件からのトレースができているか、運用をイメージして問題ないかという観点でレビューしていた。