T2R2 東京科学大学 リサーチリポジトリ
Science Tokyo Research Repository

論文 / 著書情報
Article / Book Information

題目(和文)	悪条件分析に基づく要求仕様レビュー技法の研究
Title(English)	
著者(和文)	谷﨑浩一
Author(English)	Koichi Tanizaki
出典(和文)	学位:博士（工学）, 学位授与機関:東京工業大学, 報告番号:甲第11998号, 授与年月日:2021年3月26日, 学位の種別:課程博士, 審査員:出口 弘,三宅 美博,山村 雅幸,石井 秀明,小野 功
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第11998号, Conferred date:2021/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Powered by T2R2 (Science Tokyo Research Repository)

令和二年度 博士論文

悪条件分析に基づく
要求仕様レビュー技法の研究

東京工業大学 情報理工学院

情報工学系 知能情報コース

谷﨑 浩一

概要

本研究は，人工物の設計における実装前段階での検証の方法として，製品開発における要求仕様書のレビューに着目し，境界条件のモデリングのプロセスを含む新しい要求仕様レビュー技法を提案することを目的とする．仕様の設計者が想定した境界条件を理解し，想定外の悪条件の分析を行うことで，境界条件の考慮漏れに起因する問題を実装前段階で検出可能となる．本研究で提案するレビュー技法によって，製品開発の上流工程で境界条件の想定漏れに起因する問題を指摘し，設計の改善のためのフィードバックを可能とすることを目指す．

従来のレビュー技法は欠陥検出のシナリオに基づきレビュー対象のドキュメントを読んで指摘を導出するアプローチを取る．従来手法では境界条件と悪条件の分析のプロセスが明示的でなく，境界条件の考慮漏れに起因する問題の検出が難しい．境界条件の考慮漏れに起因する問題を検出するための，レビュー技法とは別に体系化された技術として，ソフトウェアテストにおけるテスト分析の手法と安全性解析の手法が存在する．本研究では，フレームワークに基づき境界条件を分析するというテスト分析の手法のアプローチと，ガイドワードを利用して設計意図から外れる悪条件を分析するという安全性解析の手法のアプローチを要求仕様レビューに応用し，境界条件のモデリングのプロセスを含むレビュー技法を構築した．

レビュー技法の構築にあたり，要求仕様レビューの概念設計を行った．概念設計では，要求仕様レビューを「レビューアが境界条件の理解と悪条件の分析を行い，悪条件への対策案を検討して仕様書へフィードバックを行う活動」と捉え，仕様の設計者・レビューア・レビュー対象製品のユーザの関係性を定義し，概念モデルを構築した．さらに，要求仕様レビューで考慮すべき境界条件の概念モデルおよび，悪条件の概念モデルを定義し，境界条件と悪条件の関係性を明らかにした．これらの概念設計に基づき，レビュー技法のプロセスおよび各プロセスにおける入出力となる情報を定義し，「悪条件分析に基づく要求仕様レビュー技法」を構築した．要求仕様レビューで考慮すべき境界条件を分析するためのフレームワークとして Boundary Breakdown Structure を提案し，悪条件を導出する際の視点および悪条件への対策案を検討する際の観点を提案した．

2 つの具体的な事例を対象とし，提案手法を用いた仕様書レビューのケーススタディを行った．ハードウェアのみから構成される製品，ソフトウェアとハードウェアの両者から構成される製品をそれぞれ対象とし，2 つの異なるタイプの製品に提案手法を適用できることを確認した．ケーススタディで導出された悪条件を分類・抽象化することで，悪条件導出のガイドワードを構築した．ガイドワードを利用することで，レビューアが当初想定できなかった新たな悪条件を導出できることを確認した．

提案手法の有効性を検証するために 2 つの実験を行った．レビュー技法の有効性を評価するには，レビューアがレビュー技法を使い実際にレビューを実施し，その結果をもとに評価する必要がある．1 つ目の実験ではレビューの未経験者である学生 6 名を被験者とし，アドホックレビューと提案手法を用いたレビューを実施し，その結果を比較した．提案手法はレビューの未経験者でも効果を得られ，フレームワークやガイドワードを利用することで境界条件の考慮漏れに気づきやすくなり，アドホックレビューと比較して製品の改善に繋がる有効な指摘を多く導出できることを確認した．2 つ目の実験では製品検証の実務に携わるエンジニア 14 名を被験者とし，被験者を 2 グループに分けてクロスオーバー試験によって，従来手法と提案手法の効果の比較を行った．従来手法と比較して，提案手法のほうが製品の改善に繋がる重要度の高い指摘を多く導出でき，欠陥修正コストから試算したレビューの費用対効果が高いことを確認した．被験者へのアンケートおよびインタビューの回答結果から，境界条件と悪条件を分析するアプローチの有効性を多くの被験者が理解し，実務でレビュー技法を活用したいと考える被験者の数が，従来手法と提案手法で差がないことを確認した．

本研究の貢献は，要求仕様書のレビューにおいてレビューアが境界条件のモデリングを行い想定外の悪条件を分析し指摘を導出するというプロセスをレビュー技法として体系化し利用可能にした点，そのプロセスで利用する境界条件の分析のフレームワーク・悪条件導出のガイドワード・悪条件への対策案を検討する際の観点を定義し，製品の改善に繋がる重要度の高い指摘を様々な視点から導出可能にした点である．

実際の製品開発プロジェクトでの提案手法の活用を支援するソフトウェアツールの開発，教育プログラムの整備，従来のレビュー技法の活用を含めた効果的なレビューの方法の研究は今後の課題である．さらなる展望として，提案手法を仕様の設計やステークホルダーとの合意形成に応用する方法の研究を進め，製品開発の上流工程における品質の作りこみに貢献していく．

目次

第1章　序論 1
1. 背景 1
2. 本研究の目的 10
3. 本研究の方法 12
4. 本論文の構成 13
第2章　従来のレビュー技法と本研究の位置付け 14
1. 序言 14
2. 従来のレビュー技法 14
3. レビューへのテスト分析の視点の応用に関する研究 18
4. 仕様の欠陥検出の手法やリスク特定のための手法 22
5. 従来のレビュー技法の課題と本研究の位置付け 23
6. 結言 25
第3章　悪条件分析に基づく要求仕様レビュー技法の構築 27
1. 序言 27
2. 要求仕様レビューの概念モデル 27
3. 要求仕様レビューで考慮すべき境界条件の整理と Boundary Breakdown Structure の提案 29
4. 悪条件の定義と要求仕様へのフィードバックのタイプの整理 33
5. 悪条件分析に基づく要求仕様レビュー技法のプロセス 39
6. 結言 41
第4章　ケーススタディに基づく悪条件導出のガイドワードの構築 43
1. 序言 43
2. ボールペンの要求仕様書を対象としたケーススタディ 44
3. パン自動会計システムの要求仕様書を対象としたケーススタディ 50
4. 悪条件導出のガイドワードの構築 57
5. 結言 60
第5章　実験による提案手法の有効性の検証 62
1. 序言 62

2. 実験 1:レビュー未経験者による提案手法の適用実験 63
(ア) 実験の目的と概要 63
(イ) 実験方法 64
(ウ) 実験結果 66
(エ) アンケートの集計結果 70
(オ) 考察 73
3. 実験 2:クロスオーバー試験に基づく従来手法との比較実験 77
(ア) 実験の目的と概要 77
(イ) 実験方法 82
(ウ) 実験結果 86
(エ) アンケートの集計結果 103
(オ) 考察 108
(カ) 実験の妥当性 115
4. 結言 116
第6章 結論と今後の展望 118
1. 結論 118
2. 今後の展望 119
参考文献 122
謝辞 131
業績リスト 132
Appendix A パン自動会計システム要求仕様書 133
Appendix B 提案手法のフレームワークとガイドワード 147
Appendix C 実験 1 のアンケートフォーム 150
Appendix D 実験 2 のアンケートフォーム 154

第１章　序論

1. 背景

工業製品などの人工物の設計において，設計に起因する欠陥が実装後に検出されると大きな手戻りが発生することから，実装前段階で設計を検証し欠陥を早期検出することは非常に重要である．特に設計と検証が難しい工程として，開発の最上流工程である要求仕様の設計が挙げられる．要求仕様の設計は，現実世界で解決すべき問題や現実世界で提供したい価値を定義し，その実現手段を設計していく過程と言え，問題を明確に定式化することや最適解を導くことが難しい領域である．要求仕様の設計者とエンドユーザなどのステークホルダーとの間で問題状況の認識を合わせ，議論し，要求仕様の内容に対する合意形成を行っていく必要があり，ソフトシステム方法論[1]に代表されるソフトアプローチにより，人間の理解を概念モデルで表現して解を模索していく必要がある．新たなシステムや製品を設計する方法論はデザイン科学研究の領域でも議論されている[2, 3]．デザイン科学のアプローチでは，解決すべき問題を特定し，ソリューションの検討，開発を実施し，成果物の評価を行い改善していく[4, 5]．要求仕様の検証は，デザイン科学のアプローチにおける成果物の評価の前に，解決すべき問題やソリューションの妥当性を設計段階で検証する，プレ評価に該当する．プレ評価の結果を設計に対してフィードバックすることで，開発・実装に着手する前の段階で，システムや製品の改善が可能となる．適切なプレ評価を実施するための方法論が必要である．

工業製品の開発においては，品質は工程で作りこむという考え方が知られ，開発プロセスの各工程で成果物の検証を行うことで品質の高い製品の開発を可能としている[6]．ソフトウェアを含む製品の開発では，上流工程からの品質の作りこみの重要性が知られ，上流工程での検証や品質保証活動が議論されている[7]．最上流工程の成果物である要求仕様に対する検証の方法として，レビューやシミュレーションが知られる[8] ．シミュレーションを実行するためには，実行に必要な振る舞いやパラメータを洗い出し，シミュレーション実行可能な形式で仕様を記述する必要があり，記述のコストがかかる．レビューは人間が可読な形式の仕様書であれば，形式を問わず実施可能なため，多くの組織で取り入れられる．上流工程で品質を作りこむためにはレビューにより要求仕様書の妥当性を検証することが重要である．

ユーザのニーズの多様化に伴い，要求仕様を製品の実装前に想定しきることが難し

く，アジャイル開発手法を採用する組織が存在する．アジャイル開発においては，一週間から数週間の開発単位で素早く設計，開発，テストをこなし製品をリリースすることで，リリースした機能の価値を市場において検証していくことができる．しかしながら，ソフトウェア開発分析データ集 2020 [9]によるとプロジェクトで利用される開発ライフサイクルモデルは表 1-1 に示したとおりで，ウォーターフォール型が 97.6%とほとんどを占める．ソフトウェア開発分析データでデータ収集の対象となっているのはソフトウェアを含む製品かつ「アプリケーションソフト」が 9 割を占めており，業務システムの構築がほとんどではあるが，組み込みシステムにおいても類似の傾向であることが推測される．

表 1-1　利用される開発ライフサイクルモデル

開発ライフサイクルモデル	割合
ウォーターフォール	97.6%
反復型	1.8%
その他	0.7%

※ソフトウェア開発分析データ集 2020 [9] のデータを元に筆者が整理した

同じくソフトウェア開発分析データ集 2020 [9]では，開発プロジェクトの 32.5%が新規開発のプロジェクトである．新規開発のプロジェクトでは既存のシステムの改修とは異なり，ユーザの要求を新たに分析する必要があり，前述の要求定義の難しさが課題となる．ウォーターフォール型の開発では，製品開発の工程は要求分析・要件定義・概要設計・詳細設計・実装・テストと分けられ，上流工程である要求分析・要件定義において，仕様書を検証することで品質を作りこむことは重要である．アジャイル開発においては仕様書などのドキュメントの作成よりも実際に動作する製品を開発することが重要視される[10]が，実装着手前の早期段階で品質チェックを行うことで欠陥を検出する重要性は認識され，要求仕様に相当するプロダクトバックログアイテムの品質チェックを行う方法が研究されている[11, 12]．製品の欠陥を修正するコストは，製品開発プロセスの上流工程ほど小さいことが知られる．テスト工程などの下流工程で欠陥が検出された場合，誤りのある設計ドキュメントの修正，欠陥そのものの修正，修正部分のテストと影響範囲に対する確認テストが必要になり，多くの作業が発生するためである[13]．文献[14]で

は要求定義/設計工程における修正コストを 1 とすると，コーディング工程における修正コストは 5 倍，テスト工程における修正コストは 25 倍，出荷後における修正コストは 250 倍になると報告されている．また，文献[15]では，要求定義工程における修正コストを 1 とすると，設計工程における修正コストは 5 倍，コーディング工程における修正コストは 10 倍，テスト工程における修正コストは 10～20 倍，出荷後における修正コストは 200 倍になると報告されている．要求仕様書の妥当性を検証し，要求分析や要求定義の段階で欠陥を早期に検出することは，コストをかけずに欠陥の修正を行うために非常に重要である．要求仕様の設計者が想定しなかった製品の利用条件や，製品として必要な機能の定義漏れが後工程で明らかになると，大きな手戻りに繋がる．特に，新製品の開発時や，当該製品の開発経験が少ない設計者が要求定義を担当する場合にはこのような想定外が発生しやすい．想定外の問題をいかに減らせるかは要求仕様書の妥当性を検証するうえで重要な課題である．

要求仕様書の記述言語には，自然言語，UML/SysML 等のモデリング言語，VDM 等の形式仕様記述言語が存在し，プロジェクトによってさまざまな形式が取られるが，自然言語で要求仕様書を記述する組織は多い．2018 年度の「組込み/IoT に関する動向調査」調査報告書[16, 17]によると，UML/SysML 等のモデルをベースとした開発技術・開発ツールの導入目的は表 1-2 に示したとおり「品質の向上」を選択した組織が 64.9%で最も多く，次に多いのは「上流工程での検証」であり，上流工程での品質の作りこみに対するニーズの高さがうかがえる．しかし，同調査における UML/SysML 等のモデリング言語の導入状況は表 1-3 に示したとおり，「すべてのプロジェクトで導入」または「一部のプロジェクトで導入」と回答した組織が合計 39.1%であり，6 割程度の組織ではモデリング言語は導入されていない．モデルベース開発技術・開発ツールの利用の際の課題は表 1-4 に示したとおり，「モデルベース開発技術を扱える技術者が少ない」が 81.4%で圧倒的に多く，技術の難しさが導入の敷居となっている実態がうかがえる．形式仕様記述言語に関しても，数学や論理学の知識が必要になるなど導入の敷居の高さから，開発の現場で広く普及しているとは言い難い現状がある[18-20]．モデリング言語や形式記述言語を導入していない多くの組織では，要求仕様書は自然言語で記載されていると考えられる．

表 1-2　モデルベース開発技術・開発ツール利用の導入目的（上位 3 つ）

導入目的	割合
品質の向上	64.9%
上流工程での検証	46.7%
開発コストの削減	36.9%

※2018 年度 組込み/IoT に関する動向調査 調査報告書（データ編）[17]のデータを元に筆者が整理した

表 1-3　モデリング言語（UML/SysML）の導入状況

導入状況	割合
すべてのプロジェクトで導入	3.1%
一部のプロジェクトで導入	36.0%
試用中・評価中	3.5%
使用していない	43.3%
知らない・わからない	14.2%

※2018 年度 組込み/IoT に関する動向調査 調査報告書（データ編）[17]のデータを元に筆者が整理した

表 1-4　モデルベース開発技術・開発ツール利用の際の課題（上位 3 つ）

課題	割合
モデルベース開発技術を扱える技術者が少ない	81.4%
ツールの製品価格が高い	46.9%
既存ソフトウェア資産の移行が困難	38.6%

※2018 年度 組込み/IoT に関する動向調査 調査報告書（データ編）[17]のデータを元に筆者が整理した

自然言語で記述される仕様書にはあいまいさが含まれる. ソフトウェア開発分析データ集 2020 [9]によると要求仕様の明確さは, 表 1-5 に示したとおり「ややあいいまい」と「非常にあいまい」の合計が 28.4%であり, 4 分の一を超えるプロジェクトで要求仕様のあいまいさが存在している. ユーザ担当者の要求仕様関与は, 表 1-5 に示したとおり「十分に関与」と「概ね関与」の合計が 88.3%であり多くのプロジェクトでユーザ担当者が要求仕様の策定に関与している. これらの調査結果から, ユーザ担当者が要求仕様の策定に関与していても, 要求仕様があいまいなプロジェクトが存在している. ユーザ担当者による要求仕様のレビューを通過して, ユーザ担当者と設計者で合意形成を行ったとしても, 実際は要求仕様があいまいな状態のままで次の工程に進んでいるケースが存在すると考えられる. ソフトウェア開発データ白書 2018-2019 [21]では, 新規開発プロジェクトにおける要求仕様の明確さと FP 発生不具合密度の関係性について, 要求仕様が明確なほうが不具合の発生密度が低い傾向が見られるとの報告がある. あいまいな要求仕様をレビューにより検証し, 要求仕様書で考慮されていない事項を指摘してあいまいさを取り除くことは, 後工程での不具合発生を防止する効果があると言える. 自然言語で仕様を記述することによって生じるあいまいさや矛盾を防ぐには形式仕様記述言語の利用が効果的ではあるが, 前述のとおり, 仕様記述の難しさから普及が進んでいない. レビュー技術の観点で考えると, 要求仕様があいまいな場合でも, レビューアは製品のユーザの目的や想定される利用状況を考慮し, 本来定義すべき要求仕様が何かを検討して, 仕様の不足や誤りを指摘する必要がある. 要求仕様書のあいまいさに関わらず, 効果を発揮できるレビュー技術が必要である.

表 1-5　要求仕様の明確さとユーザ担当者の要求仕様関与

回答	要求仕様の明確さの割合	ユーザ担当者の要求仕様関与の割合
非常に明確 / 十分に関与	26.7%	37.5%
かなり明確 / 概ね関与	45.0%	50.8%
ややあいまい / 関与が不十分	25.1%	9.5%
非常にあいまい / 未関与	3.1%	2.2%

※ソフトウェア開発分析データ集 2020 [9]のデータを元に筆者が整理した

ISO/IEC20246:2017 [22]に定義されたレビューのプロセスは，図 1-1 に示したとおりレビュー計画，レビュー開始，個人レビュー，問題点の議論および分析，修正および報告に分かれる．レビュー計画では，レビューの目的やレビュー対象とする成果物，使用する技法や参加者とその役割を特定する．レビュー開始ではレビューに必要な資料がレビューアへ配布され，リーダーからレビューアへレビューの範囲やレビューアの役割，レビュー時に考慮すべきポイントを通知する．個人レビューではレビューアがレビュー技法を用いて仕様書への指摘を導出する．問題点の議論および分析では，導出された指摘に基づく仕様書の修正の要否を議論し，レビュー対象を修正するか修正せずに次の工程に進むかといった決定を行う．修正および報告では，指摘に基づいて仕様書が修正されたのちに修正結果が報告される．

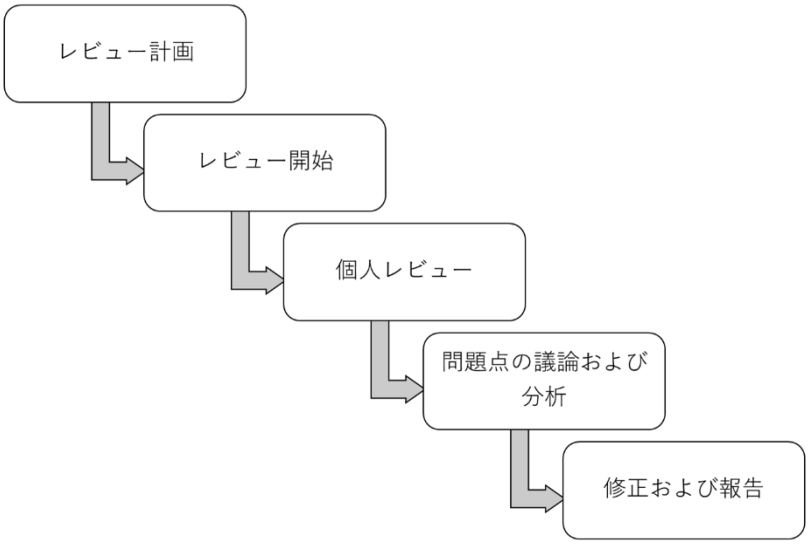


図 1-1　ISO/IEC20246:2017 におけるレビューのプロセス
※ISO/IEC20246:2017[22]の Figure 1 を日本語訳して引用

仕様書のレビューは，仕様に基づき開発された製品の振る舞いを机上でシミュレーションし，仕様の欠陥を検出する行為である[23]．レビュー対象の構造や振る舞いに関する認識モデルに基づき，レビューアは机上でシミュレーションを行う．製品，利用者，利用環境の相互作用を想定した様々なシミュレーションのシナリオを机上で実行するには，仕様書の内容をレビューアが理解し認識モデルを構築することが必要である．レビューアは，製品の振る舞いやユーザによる製品の利用方法を想定し，机上でのシミュレーションで様々なシナリオを実行することで，各シナリオで発生する欠陥を見出し，指摘

事項を導出する. レビューアの認識モデルの構築にあたっては, 製品自体の機能, 利用者, 利用環境などの様々なアスペクトから境界条件を分析することが必要である. 様々なアスペクトから境界条件を分析することで, レビューアは多様なシナリオでシミュレーションを実行でき, 欠陥をより多く検出できる.

対象のモデルの策定時に取り扱われる変数は, 所与のものとして外的に与えられる変数と内的に決定できる変数とに分けられる. 一般的に, 所与のものとして外的に与えられる変数は, モデルの境界条件と呼ばれる. 本研究における「境界条件」は, レビュー対象の製品の機能に影響を与える外的条件を表し, レビュー対象の製品の仕様を決定するために仕様の設計者やレビューアによって想定される. 仕様を決定するために考慮が必要な, 製品のユーザ・ユースケース・利用環境などの, 内的に決定できない所与の条件が境界条件にあたる. ソフトウェアがレビュー対象の場合, 製品外部の条件だけでなく, 製品内部のハードウェアに関連する条件もソフトウェアにとっては所与の外的条件と考えられる. スマートフォンのアプリケーションを例にすると, 動作保証する OS のバージョンや画面サイズ, ユーザがスマートフォンを持つ向き, 周囲の明るさや環境音の大きさなどが境界条件として考えられる. 要求仕様書で考慮が漏れた外的条件はそれ以降の工程でも考慮されないまま製品開発が進み, 製品をリリースした後で想定外の条件で製品が利用され, 不具合が発生するなどして問題となるケースがある. 想定外の悪条件に起因する仕様の欠陥を検出することは, 要求仕様書のレビューにおける重要な課題と言える. 本論文における「悪条件」は, 仕様の設計者の想定範囲を超える外的条件を表す. 境界条件として想定された所与の外的条件の範囲を超える条件が, 悪条件である. 悪条件によって, 所与の外的条件を想定して設計された製品の仕様が満たされなくなる. 仕様書のレビューの際に悪条件を想定して仕様の欠陥を指摘することで, 悪条件への対策案を仕様に盛り込むことが重要である.

レビューのプロセスの中で, レビューアが境界条件を想定し机上シミュレーションを行う主要なプロセスは個人レビューである. 問題点の議論および分析においても, 議論の中で新たな境界条件が見出される可能性はあるが, 数名の参加者による議論を行う場で境界条件を一から整理するのは時間がかかり効率が悪い. 個人レビューで事前に境界条件を整理し, その結果を元に議論を行うのが効率的と考えられる. 個人レビューの技法として, ISO/IEC20246:2017 [22]では Ad hoc reviewing（アドホックレビュー）, Checklist-based reviewing (CBR), Scenario-based reviewing (SBR), Perspective-based

reading (PBR) , Role-based reviewing (RBR)が挙げられている. PBR は様々なステークホルダーの視点からレビュー観点を設定することで, 網羅的な欠陥の検出が可能となる[24, 25]. これらの従来のレビュー技法はチェックリストやレビュー観点の設定方法に着目しているが, 境界条件の分析に関する議論はされていない. 文献[26]では, 様々な境界条件のもとでのシステムの振る舞いを机上でシミュレーションするにあたって, システムの仕様をマクロに見て設計意図を満たしているかという視点から評価するだけでなく, ミクロな視点から境界条件を把握することの重要性が述べられている. ミクロな視点から把握した境界条件を構造化し, 仕様書で考慮が漏れた境界条件を補完してモデリングする方法論が求められる.

境界条件の整理の方法を体系化し検証に利用している事例として, ソフトウェアテスト分野におけるテスト分析という工程がある. テスト分析では, テスト対象であるソフトウェア製品の機能やユーザによる利用状況を理解し, テストにおいて考慮すべき境界条件を構造的に整理する. 様々な分析のフレームワークや境界条件を整理するための記法を用い, 仕様書に記載されていない境界条件まで想定範囲を広げて整理する方法がいくつか提案されている[27-30]. テスト担当者が要求仕様をインプットにし, 境界条件を整理しながらテスト設計を行うと, 要求仕様の不備が検出されることが多々ある. テスト担当者による要求仕様の不備の検出をプロジェクトの早期段階で実現し欠陥の未然防止に繋げるために, テスト担当者が要求定義工程に関わることの重要性が古くから議論されている. そのための実践的な施策の一つとして要求仕様レビューへのテスト担当者の参加が挙げられる[31-33]. 要求仕様レビューへテスト担当者が参加するメリットとして, 要求の欠陥と漏れの摘出が期待できる一方, デメリットとしては間違った視点からの指摘や詳細すぎる指摘の増加が懸念される[30]. 単にテスト担当者を要求仕様レビューに参加させれば良いわけではなく, テスト分析における境界条件を整理するプロセスやフレームワークを要求仕様レビューに適した形に構成し, レビューアが利用可能なレビュー技法として体系化する必要がある.

要求仕様レビューは人間の思考に基づいて実施される行為であり, 実施結果は個々の人間に依存する. 境界条件の整理のプロセスをもうけ, 何らかの分析のためのフレームワークを利用したとしても, 一人の人間がありとあらゆる境界条件を想定し要求仕様書の問題点を全て指摘することは, 現実的には困難である. 文献[34]では, 製品のユーザビリティの評価に的を絞った実験結果ではあるが, 一人の評価者が見つけられ

る問題点は全体の 35%しかなく，最低 3 人の異なる評価者が評価を実施することで妥当な結果が得られると報告されている．複数の独立したチームがそれぞれレビューを行うことで多くの欠陥を検出する方法として N-Fold Inspection が提案されている．1 チーム 4 名で 10 チームを被験者としたパイロットスタディの結果では，特定の指摘を見つけられたチームの数は平均で 4 チーム以下であると報告されている[35]，レビューアの人数やチームの数を増やすほど見つけられる指摘は増えるが，その分レビューにかかるコストが大きくなるため，プロジェクトのリソースの制約上，レビューアの人数やチームの数を十分確保することが難しい場合もある．レビューの効果を高めるには，一人ひとりのレビューアが思考できる境界条件の範囲を広げるアプローチが重要となる．

以上を整理すると，製品の要求仕様書を適切に検証するためには，(1)要求仕様書が自然言語であいまいさを含み，境界条件が不明瞭な場合でも，レビューの効果を発揮でき，(2)境界条件の整理と想定外の悪条件の検討のプロセスを陽に含み，(3)要求仕様レビューに適した共計条件の分析のフレームワークを持ち，(4)境界条件の想定範囲を広げるアプローチによる，個人レビュー技法が必要であると言える．本研究ではこれらの要件を満たす個人レビュー技法を提案し，提案手法の効果の検証を行う．実際の製品開発プロジェクトでは，要求仕様書のレビューで指摘が導出されると，指摘は要求仕様書作成のプロセスにフィードバックされ，要求仕様書の修正と再レビューを経て合意形成に至るプロセスが存在するが，本研究ではレビューにおいて指摘を導出する行為に焦点をあてることとし，要求仕様書の修正や合意形成のプロセスは対象外とする．図 1-2 に本研究の対象範囲を示す．個人レビューにおいて，要求仕様書を理解し境界条件を整理・モデル化し，想定外の悪条件を検討して仕様書への指摘を導出する行為が本研究の対象範囲である．

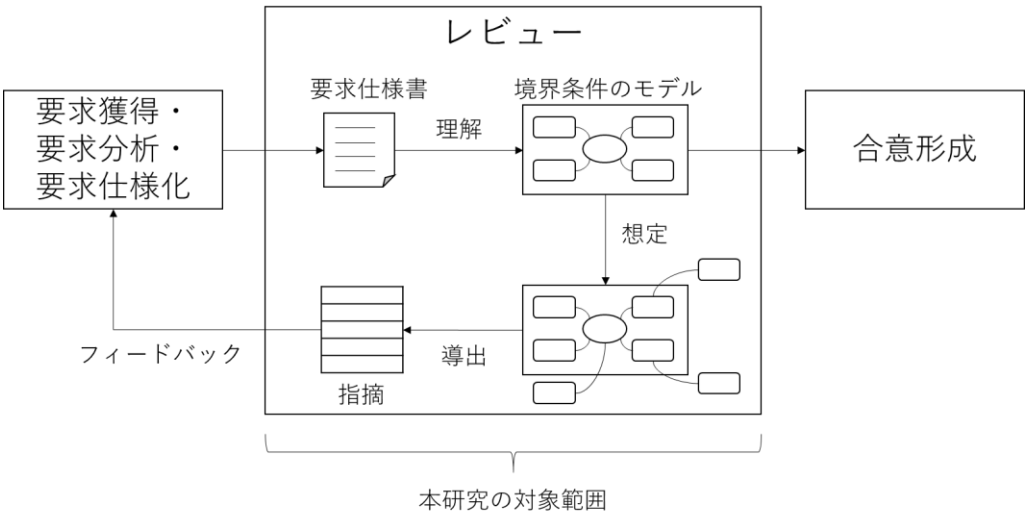


図 1-2　本研究の対象範囲

2. 本研究の目的

本研究では，人工物の設計におけるプレ評価の方法論を構築することで，開発の上流工程で境界条件の想定漏れに起因する欠陥を検出し，設計の改善のためのフィードバックを可能とすることを目指す．そのために，個人レビューのレビュー技法を研究対象とし，前述の 4 つの要件を満たす要求仕様レビュー技法の提案を目的とする．本研究の目的は次の 3 つとする．

目的 1：境界条件の整理のプロセスを含むレビュー技法の構築

要求仕様レビューの概念を整理し，レビュー技法のプロセスを定義する．さらに，要求仕様レビューで考慮すべき境界条件の構造を明らかにし，境界条件を整理するためのフレームワークを構築する．要求仕様レビューの概念の整理では，要求仕様レビューに関連するステークホルダーと，レビューの際に考慮すべき概念の関係性を定義する．これにより境界条件の整理のプロセスを陽に含むレビュー技法および，境界条件の整理のフレームワークが構築され，前述の要件(2)と要件(3)を満たす．

目的 2：悪条件の考慮範囲を広げるためのガイドワードの構築

ケーススタディにより提案手法の事例への適用を示すとともに，ケーススタディで導出された悪条件を分類・抽象化することで，レビューアが想定外の悪条件を検討するため

のガイドワードを構築する. ガイドワードを用いてレビューアが悪条件を検討することで, 境界条件の想定範囲を広げることが可能となり, 要求仕様があいまいな場合でもレビューアが広い範囲の境界条件を考慮して悪条件を導出できる. これにより前述の要件(1)と要件(4)を満たす.

目的 3: 提案手法の有効性の検証

提案手法の有効性を検証するために, レビュー技法単体として提案手法のアプローチが正しく機能することの検証, および従来のレビュー技法と比較した際の提案手法の有効性の検証, という 2 つの検証を行う. 前者の検証では, 提案手法のプロセスとフレームワークを利用して, 設計者の想定外の悪条件に起因する仕様の問題点をレビューアが指摘可能か確認する. また提案手法を用いない場合に比べてレビューアの考慮範囲が広がるか確認する. 後者の検証では, 従来のレビュー技法と比較して提案手法のほうが, 製品の改善に繋がる重要度の高い指摘を多く導出できるかを確認し, 欠陥修正コストから試算したレビューの費用対効果を確認する. また仕様書のあいまいさに関わらずレビューアの考慮範囲が広がるか, レビューア自身が提案手法の効果をどのように感じたかを主観評価により確認する. これらの検証により, 提案手法が要求仕様レビューに有効であることを実証する.

以上の目的を整理し, 本研究の目的の全体像を図 1-3 に示す. これらの目的を満たすことで, 前述の 4 つの要件をそなえた有効なレビュー技法を提案する. 従来のシナリオベースのレビュー技法では, 要求仕様とチェックリストやレビュー観点を突き合わせてレビューを行い, 境界条件の整理のプロセスとフレームワークとガイドワードが存在しないため, 想定外の悪条件に起因する欠陥を検出することが難しい. またレビューアが認識した境界条件が明確にならないため, レビュー指摘をもとに設計者とレビューアで議論する際に, なぜその指摘が重要であるかが自明でなく議論が難しい. 本研究で提案するレビュー技法を使用することで, レビューアは所与の境界条件の把握と悪条件の検討を明示的に実施でき, 設計者との想定範囲のギャップについて議論することができるようになる. 仕様の設計・レビューの時点でこの議論を実施できると, 大きな手戻りコストに繋がるような想定外の悪条件に起因する欠陥が後工程で見つかるリスクや, 製品を市場リリースした後で想定外の悪条件に起因する欠陥が見つかり大きな改修コストが

発生するリスクを低減することが可能となる.

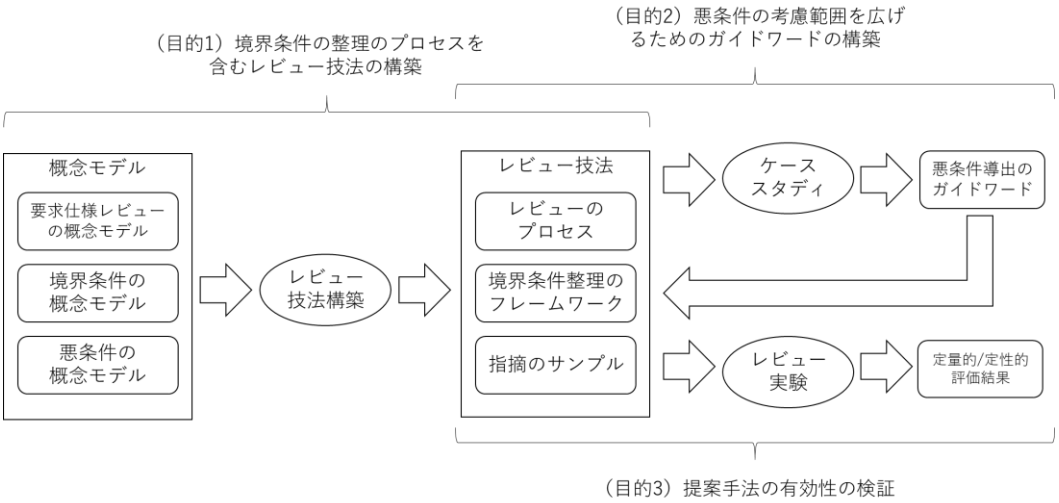


図 1-3　本研究の目的の全体像

3. 本研究の方法

レビュー技法を構築するにあたり，要求仕様レビューの概念モデルをおよび悪条件の概念を定義する. 要求仕様のレビューで考慮すべき境界条件の構造を明らかにし，レビューのためのフレームワークを構築する. 概念モデルとフレームワークをベースに，境界条件のモデリングを含むレビューのプロセスを設計する.

次に，フレームワークとレビューのプロセスの事例への適用性を示すために，ケーススタディを行う. ケーススタディでは事例への適用性を確認するとともに，レビューアが悪条件を導出する際の観点を明らかにする. ケーススタディで導出された悪条件を分類・抽象化し，悪条件を導出するためガイドワードを構築する.

最後に，提案手法を使ったレビューの実験を行い，提案手法の有効性を確認する. レビュー技法の効果を評価するには，実際に人間がレビュー技法を使いレビューを実施する必要がある. まず提案手法単体として，本研究で提案する境界条件と悪条件の分析のアプローチが正しく機能し，レビューアの考慮範囲が広がり有効な指摘を導出できることを検証する. レビューの未経験者に提案手法を用いたレビューを実施してもらい，製品の改善に繋がる有効な指摘を導出できるかを確認する. 次に，従来のレビュー技法と比較した際の提案手法の有効性を検証する. 実務に携わるエンジニアに従来手法によるレビューと提案手法によるレビューを実施してもらい，製品の改善に繋がる

重要度の高い指摘を多く導出できるか，欠陥修正コストから試算したレビューの費用対効果が高いかを確認する．

4. 本論文の構成

本論文の構成は次の通りである．第 2 章では，従来のレビュー技法について解説するとともに，境界条件の考慮漏れに起因する問題の検出を行っている検証手法として，ソフトウェアテストにおけるテスト分析の手法や安全性解析の手法に関する先行研究を紹介し，従来のレビュー技法の課題と本研究の位置付けを明らかにする．第 3 章では，要求仕様レビューの概念モデルを定義し，要求仕様レビューで考慮すべき境界条件の構造および悪条件の概念の定義を行ったたうえで，悪条件分析に基づく要求仕様レビュー技法を構築する．第 4 章では提案手法を事例に適用したケーススタディを示し，ケーススタディにて導出された悪条件を分類・抽象化し，悪条件を導出するためのガイドワードを構築する．第 5 章では提案手法の有効性を検証するためのレビューの実験について，目的と概要・方法・結果・考察を述べる．第 6 章で結論と今後の課題を述べる．

第２章　従来のレビュー技法と本研究の位置付け

1. 序言

本章では，先行研究として，従来のレビュー技法に関する研究や複数のレビュー技法の効果の比較に関する研究を紹介する．従来のレビュー技法は，欠陥検出のシナリオに基づきレビュー対象のドキュメントを読み指摘を導出するアプローチであり，境界条件と悪条件の分析のプロセスが明示的でなく，境界条件の考慮漏れに起因する問題の検出が難しいことが課題である．境界条件の考慮漏れに起因する問題の検出を行っている検証手法として，ソフトウェアテストにおけるテスト分析の手法や安全性解析の手法に関する先行研究を紹介し，フレームワークに基づき境界条件を整理するというテスト分析のアプローチや，ガイドワードを利用して設計意図から外れる悪条件を分析するという安全性解析手法のアプローチを，要求仕様レビューに応用する必要性を論じる．

2. 従来のレビュー技法

システム工学分野におけるレビューの国際規格である ISO/IEC20246:2017 [22]では，個人レビューのレビュー技法として，Ad hoc reviewing（アドホックレビュー），Checklist-based reviewing (CBR)，Scenario-based reviewing (SBR)，Perspective-based reading (PBR)，Role-based reviewing (RBR)が挙げられている．その他のレビュー技法として，SQuBOK Review2017 [36]では Defect based-reading（DBR）と Usage-based reading（UBR）が挙げられている．

アドホックレビューは，構造化された方法やプロセスのない非公式なレビューである[22]．レビューアが自由にレビュー対象の文書を読み，気になった点や問題点を指摘する．手軽に実施できる反面，レビューの結果がレビューアの能力に大きく依存する[25]．どのような視点でレビューを行うかはレビューア任せとなるため，導出した指摘に対してなぜその指摘が必要だと考えたかを第三者に説明しレビュー結果の妥当性を示すことや，組織的にレビューアのレビュー能力を向上させる教育を行うことが難しい．

CBR は，レビューアがチェックリストに基づいてレビュー対象の文書を読み欠陥を検出する手法である．チェックリストは過去の経験などに基づき，組織やプロジェクトごとに作成される．文献[37]では，個人レビューのレビュー技法を利用する組織のうち 50%がCBRを利用していると報告されており，広く使われるレビュー技法である．効果的なレビ

ューを行うためには適切なチェックリストを用意し，チェックリスト自体をプロジェクトの状況に応じて適切に更新していく必要がある．チェックリストに記載されている範囲では効率的に指摘を見つけることができる反面，チェックリストの範囲を超える指摘にレビューアが気付きにくくなるという課題がある．チェックリストが適切に更新されず，レビューアが何も考えずにチェックリストの範囲だけでレビューを実施すると，有効なレビューを実施できないリスクがある．

SBR は欠陥検出のためのシナリオを作成し，シナリオに基づいてレビュー対象文書を読み欠陥を検出する手法である．開発する製品のユースケースをシナリオとして使用する方法や，検出したい欠陥の種類を定めその種類の欠陥を検出するためのシナリオを作成する方法がある．検出したい欠陥の種類を定めてシナリオを作成してレビューする方法は DBR と呼ばれる．レビューにおけるシナリオ作成の重要性は文献[38]でも言及され，システムが提供する価値とその価値を阻害する要因を想定して，検出したい欠陥の種類を特定する手法が解説されている．システムが提供する価値とその価値を阻害する要因を特定する方法は明示的でなく，仕様を理解してそれらを想定する思考プロセスの明確化が必要である．検出したい欠陥の種類を特定する方法は，システムが提供する価値から想定するアプローチのほかに，旧バージョンの開発プロジェクトや類似のプロジェクトにおけるレビューでの指摘やテストで見つかった不具合の情報を元に検討するアプローチがある[13, 39,40]．過去に発生した欠陥と同様の欠陥が作りこまれることを防止するのに役立つが，想定外の欠陥の検出は難しい．また新製品の開発の際など，過去の欠陥情報が存在しない場合にこのアプローチは利用できない．SBR の課題として，チェックリストと同様に，規定されたシナリオの範囲を超える指摘にレビューアが気付きにくくなるという課題がある．

欠陥情報そのものではなく，欠陥を埋め込む人間の認知プロセスに着目し，欠陥の作りこみに繋がる罠（トラップ）を想定してレビューに利用する方法として Trap-based Review が提案されている[41]．Trap-based Review では，技術者が欠陥を作りこむ要因となる罠をトラップと呼び，トラップに基づいて仕様の問題点を検出するためのシナリオを作成してレビューを行う．DBR と同じく，SBR の一種と考えられる．Trap-based Review の手順にはトラップをベースに欠陥検出のシナリオを作成するプロセスは示されているが，レビュー対象そのものの具体的な境界条件を分析するプロセスは明示的ではない．トラップに着目してレビューすることで設計者の考慮漏れや想定外の問題を指摘するこ

とは可能だが，SBR の課題である，シナリオに記載された以外の問題点に気付きにくくなるという点は Trap-based Review にも当てはまる．

PBR と RBR は複数のステークホルダーの視点（パースペクティブ）からレビューを行う手法である[42]．ビジネスアナリスト，ビジネスオーナー，保守担当者，マーケティング担当者，オペレータ，プログラマなど様々なステークホルダーのパースペクティブを利用できるが，文献[42]や文献[43]で利用されているように，PBR では一般的には利用者，設計者，テスト担当者の 3 つのパースペクティブが知られる．レビューを実施する前に，パースペクティブごとに欠陥検出のためのシナリオを用意する．レビューアは特定のパースペクティブを割り当てられ，自身が担当するパースペクティブのシナリオを用いてレビュー対象の文書を読むことで欠陥を検出する．様々なステークホルダーの視点でレビューを行うことで，レビューの網羅度を上げることができる．RBR では日常のレビューア自身の役割とは異なる役割のパースペクティブからレビューを行うことで，普段気付かない指摘に気付くことができる．レビューアごとにパースペクティブを分担することで，指摘の重複を防ぐ効果も期待できる．CBR や SBR と同様，シナリオに規定された範囲を超える欠陥にレビューアが気付きにくい点が課題である．

UBR はユースケースを元にレビュー対象の文書を読み，欠陥を検出する手法である[44]．ユースケースに優先順を決め，優先順の高いユースケースからレビューを行う．ユースケースとは，利用者が製品やシステムを用いて特定の目的を達するまでの，製品やシステムとのやり取りを表す．UBR ではユースケースの流れを追いながらレビュー対象の文書を読むことで，必要な機能の漏れなどの問題点を指摘する．ユースケースは複数の境界条件を含み，ユースケースに基づいてレビュー対象の文書を読み製品やシステムの振る舞いを想定することは，ユースケースを机上シミュレーションのシナリオとして用いて机上シミュレーションを実行する行為と言える．すべてのユースケースをレビューに利用し終わるか，事前に定めた時間になったらレビューを終えるため，レビューの終了規準は分かりやすい．ユースケースが明確な場合に，ユースケースを達成するのに最低限必要な機能が備わっているかを確認することは可能だが，想定外の利用方法や利用環境など，ユースケースに含まれない外的条件に関しては別途考慮する必要がある．

SQuBOK Review2017 [36]では PBR・CBR・DBR はシナリオベースのレビュー技法に分類される．これらのレビュー技法はいずれも欠陥検出のためのチェックリストやシナリ

オを用意し，チェックリストやシナリオに基づいて仕様書を読んで欠陥を指摘するというアプローチを取る．これらのレビュー技法は課題も共通しており，チェックリストやシナリオに規定された範囲を超える欠陥にレビューアが気付きにくくなる点が課題として挙げられる．与えられたチェックリストやシナリオの範囲にレビューアの思考が限定されることで，欠陥の見逃しに繋がるリスクがある．この課題に対処するには，レビューア自らが仕様で考慮された境界条件を理解して，仕様の設計者の想定外の悪条件まで考慮範囲を広げるアプローチが必要となる．

シナリオベースのレビュー技法である PBR・CBR・DBR とアドホックレビューの効果の比較に関する研究がいくつか存在する．文献[42, 45]はアドホックレビューと PBR の比較を行い，PBR が有意との結果が出ている．文献[46, 47]は PBR と CBR の比較を行い，PBR は有意でないとの結果が出ている．文献[48-51]は DBR の有意性について CBR やアドホックレビューとの比較を行っているが，有意な結果と有意でない結果がある．文献によってどの手法が優位かの実験結果が異なっており，一概にどの技法が有効と断定することは難しい．この理由として，チェックリストやシナリオをベースとしたレビュー技法は，レビューの結果がチェックリストやシナリオに依存しやすく，レビュー対象に合った適切なチェックリストやシナリオが設定されないと効果的なレビューを実施するのが難しいことが考えられる．

レビュー技術の研究としては，1990 年代から 2000 年代前半にかけて前述のレビュー技法に関する議論が行われてきたが，2010 年代以降は，DBR のアプローチに基づき重点的に検出すべき欠陥を明らかにしてレビューを行う方法や，ツールを利用して非同期・遠隔でレビューを実施する方法など，効率よくレビューを行う方法に関する研究が注目されてきた．前者については，手戻りコストが大きい欠陥を検出するレビュー方法[13]，製品・システム特有の欠陥の検出を熟練者が担当するレビュー方法[39]，欠陥情報のリポジトリからレビューシナリオを作成する方法[40]，上流工程の文書から下流工程の文書を作成した際に漏れたキーワードを半自動検出する方法[52]が提案されている．後者はモダンコードレビューと呼ばれ，文献[53]で紹介された CodeFlow や GitHub[54]などのツールを用いて非同期・遠隔で効率的にソフトウェアのソースコードをレビューする方法であり，前述のレビュー技法のような欠陥の検出方法に関する技術とは異なる．レビュー技術の研究において，境界条件を整理して想定外の悪条件まで考慮範囲を広げるアプローチは議論されてこなかった．

要求仕様書で定義が漏れている境界条件や想定外の悪条件を考慮してレビューするには，従来のレビュー技法のような，所与のチェックリスト・シナリオ・ユースケースの範囲内で指摘を探すアプローチでは不十分である．要求仕様書で考慮すべき境界条件をレビューア自らが整理し，その想定範囲を広げるように思考するアプローチを取ることで，個々のレビューアが考慮できる範囲が広がり，仕様の設計者が想定しなかった悪条件に起因する仕様の問題を検出できる．

3. レビューへのテスト分析の視点の応用に関する研究

ソフトウェアテストではレビューは静的テストの手法の一つに分類される．レビューは実際に製品を動作させて実施する動的テストより前に欠陥を検出するための活動として広く実施される．ソフトウェアテストの国際資格団体であるISTQBによる調査 [55]では，回答者の約 70%が動的テスト実行の前に要求仕様書のレビューを実施しており，要求仕様書のレビューは欠陥の早期検出のための最も広く実施される活動であるとしている．この調査の回答者は半数以上がテスト担当者やテストリーダによるものであり，テストに関わるエンジニアの多くが要求仕様書のレビューを実施していると考えられる．

テストの観点を要求仕様書のレビューに応用することの重要性や実践のための方法論の必要性が認識されている[31-33]．テストの設計では，テスト対象を分析しテストの内容を整理するテスト分析というプロセスが存在し，テストで考慮すべき境界条件を整理していく過程で仕様書の欠陥を検出することがある．テストの視点で仕様書を分析することが，一種のレビューの効果を生んでいると考えられる．

テスト分析の方法論として，VSTeP[27]，HAYST 法[28]，テストカテゴリや欠陥知識に基づくテスト分析法[29]，脆弱性・悪条件を考慮したテスト項目設計法[30]などが提案されている．

VSTePではNGTというツリー構造の記法を提案しており，テストで考慮すべき観点をNGT の記法に沿って整理する．テスト観点間の関連を自由に定義することで，複数のテスト観点を組み合わせてテストの内容を表現することができる．テスト観点の導出はトップダウンとボトムアップで行われるとされ，抽象化と具体化を繰り返し行うことで，考慮範囲を広げてテスト観点を整理することができる[27]．

HAYST 法は 6W2H の視点でテスト対象を分析する．6W2H とは，開発の視点としての Why（要求）・What（仕様）・How to（設計），Why のサブ視点としての Whom（お客様

のお客様を満足させる要因)・How Much(価格・販売数の予測), 顧客の視点としての When(いつ)・Where(どこ)・Who(だれ)を表す. 分析結果は 6W2H の視点でツリー構造にブレークダウンして整理する. そのうえで, FV 表と呼ばれる表形式で, テスト対象の機能を目的の粒度で分割した「目的機能」に対して, テスト条件を検討していく. 6W2H のうちユーザの視点である When, Where, Who を意識して, ユーザが製品を使用する目的の単位で製品の機能を整理しなおすことで, ユーザの目的達成の可否を検証することができる[28].

テストカテゴリに基づくテスト分析法では, 機能の論理構造モデル[56]に基づくテスト対象の機能の分割方法および, テストデータの入出力パターン[57]に基づき網羅的にテストケースを作成する手法が提案されている. 論理構造モデルやテストデータの入出力パターンを分析のフレームワークとして用いることで, 分析を行うエンジニアごとの分析結果のばらつきを抑えることができる. テストカテゴリは, テスト対象の製品ドメインに固有の具体化されたテストタイプであり, テスト対象に関する知識から導出される. テストカテゴリをプロジェクトのメンバーで共有し, ステークホルダーと合意形成することで, テストを検討する範囲を特定できる. 分割したテスト対象の機能はフィーチャ一覧として階層構造で整理し, フィーチャ一覧を縦方向にテストカテゴリの一覧を横方向に並べたマトリクスを生成し, テストすべき内容を網羅的に特定する[58].

脆弱性・悪条件を考慮したテスト項目設計法では, ソフトウェアの構造や作りに基因する弱さを脆弱性, ユーザによる想定外の製品の利用方法や想定外の利用環境などの外的条件を悪条件と呼び, テスト対象の構造と振る舞いの分析結果から, 脆弱性と悪条件を導出してテスト項目を作成する. テスト対象の構造はテスト対象構造図という構造モデルを用いて分析し, テスト対象の機能をツリー構造で整理する. テスト対象の振る舞いは, シーケンス図による機能間の相互作用の分析と, フローチャート形式の図によるユーザの製品操作のフローの分析を行う. テスト対象そのものの機能のみならず利用者や外部環境を含め, テストで考慮すべき境界条件を整理し, 様々な視点から悪条件を想定することで, 欠陥検出に有効なテスト項目を導出できる[30].

これらの手法は分析の着眼点やフレームワークや分析結果の表現方法に違いはあるが, 仕様の内容を分析しなおしテストで考慮すべき境界条件をモデリングする点が共通している. テスト担当者がテスト分析を行っている際に, 仕様の考慮漏れや仕様の設計者が想定していなかった外的条件に気付き指摘できるのは, このような分析のアプロ

ーチによるものと考えられる. テスト分析のアプローチのレビューへの応用として, 境界条件を整理してレビュー観点を設定する方法[59]や, レビューシナリオの設計の重要性が議論されている[60]が, 境界条件の分析に基づいて想定外の悪条件を考慮して仕様への指摘を導出する方法は体系化されていない.

悪条件を考慮したテスト分析の概念を図 2-1 に示す. テスト対象である製品やソフトウェアを理解し考慮すべき境界条件を明確にしたうえで, 製品の役割や機能の達成を阻害する悪条件を検討することで, 設計者や開発者の想定外の欠陥を検出可能なテスト項目を作成できる. このようなテスト分析のアプローチを取り入れたレビュー技法を構築することで, テスト担当者がテストの活動のなかで行っている, 仕様の考慮漏れに対する指摘の導出や想定外の悪条件を考慮した仕様の改善案の提案を, レビューで実施可能にする必要がある.

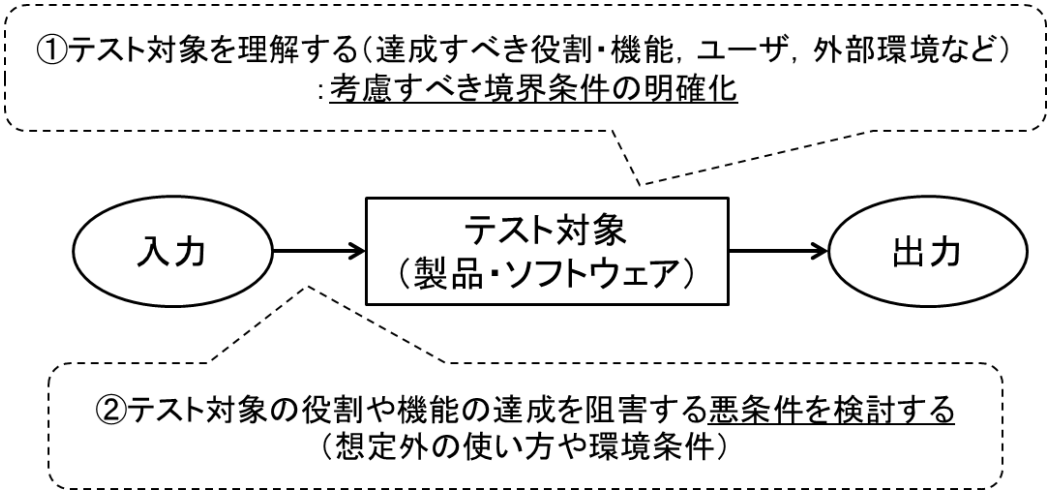


図 2-1　悪条件を考慮したテスト分析

テスト分析の成果物の表記方法としてツリー構造が採用されるものが多い. VSTeP における成果物の表現記法である NGT や, HAYST 法における分析のフレームワークである 6W2H はいずれもツリー構造の記法を取る. 6W2H のような規定のノードに対して子要素を検討することで境界条件の考慮漏れを防いだり, 境界条件の親子関係をとらえて構造化することで構造のなかに不足する要素に気付きやすくなり, 仕様書に記載のない境界条件まで想定範囲を広げることが可能となる. テストカテゴリに基づくテスト分析法や脆弱性・悪条件を考慮したテスト設計法においてもテスト対象の機能を一覧

化する際に階層構造で表現している．システムのディペンダビリティをステークホルダーに共有し合意形成するために利用される D-Case では，Goal Structuring Notation というツリー構造の記法を採用している[61, 62]．考慮すべき境界条件を論理的に展開して整理するには，ツリー構造で要素をブレークダウンしていく記法が利用しやすいと考えられる．

PBR では，パースペクティブの一つとしてテスト担当者の視点を含めることが提案されており，「テスト仕様，テスト計画，テストケースを用意するのに十分な情報がそろっているか」「検証可能（テスト可能）な要件になっているか」といった観点で仕様書を読むことで指摘を導出する[63]．文献[25]ではテスト担当者の視点から指摘を導出する手順として，同値パーティションを構築してテストケースを検討することで要求仕様の欠陥を指摘する方法が解説されている．仕様を元に同値パーティションを構築する方法は同値分割法というテスト技法の一つとして知られる[64]．同値分割法は，テストの結果が同じになると想定される入力条件を同一のパーティションに振り分ける技法である[65]．同値パーティションを構築することでレビュー対象の仕様書からテスト項目を具体的に検討することになるため，入出力の定義不足や一般的な入出力のパターンと異なるといった問題点の指摘は可能である．しかし，同値パーティションに振り分けるべき様々な入力条件を想定するための観点や，考慮すべき境界条件を分析してモデリングするというテスト分析のアプローチは示されていない．同値パーティションを網羅的に洗い出しレビューを行うためには，境界条件のモデリングによって考慮すべき境界条件を分析する必要がある．

要求仕様書におけるテストエンジニアの視点を活かした欠陥検出方法[66]では，要求仕様書を元に要求仕様書分析表とユースケースドキュメントを作成することで仕様を分析し，仕様への指摘を導出する．文献[47]では，要求仕様書分析表を用いることで，分析表を使わずに仕様書を読んだだけの場合よりも多くの欠陥を検出できたとの報告がある．このことから，要求仕様書を分析し，整理しなおすテスト分析のアプローチが仕様書レビューにも有効であることが分かる．文献[66]では，要求分析表に記載する要素として次の 11 項目が提案されている：(1)製品に求められる機能要件，(2)製品に求められる非機能要件，(3)ユーザが製品を使用する目的，(4)製品に対して入力を与えるアクタ，(5)製品が動作する状況，(6)製品が動作する契機となるイベント，(7)製品に対する入力，(8)状況・イベント・入力にもとづく製品の動作内容，(9)製品に期待される出力や

動作結果, (10)製品からの出力を受け取るアクタ, (11)不具合発生の影響から判断した重要度の分類. レビューアは要求仕様書を要求仕様書分析表の要素に分解していくなかで, 仕様の問題点に気づき指摘を導出する. この手法では, レビュー対象と関連する外的条件となりうる要素として, 製品が動作する状況が挙げられているが,「状況」には様々な要素が考えられ, 境界条件を明確にするためには「状況」の要素をブレークダウンして整理する必要がある. また, 要求仕様書分析表から指摘を導出するプロセスは定義されておらず, 要求仕様書分析表から指摘を導出する際の思考プロセスは不明確である. 仕様を分析した結果から指摘に至るまでの思考プロセスを明確にする必要がある.

4. 仕様の欠陥検出の手法やリスク特定のための手法

Quality Inspection と呼ばれる品質検査および欠陥予防の手法では, 要件定義書等の文書に対して, 定量的な情報を元に欠陥が存在しそうな箇所を推定してレビューを実施する[67]. 仮説駆動型レビュー手法も同様のアプローチを取り, 仕様の欠陥につながりうる兆候に基づきその兆候に至った背景を仮説として検討し, 仮説に基づき発生しうる欠陥を予測して欠陥検出のシナリオとする[68]. これらの手法では, 句読点の数や文のコピー率, ファイルの更新日時といった特徴量を欠陥に繋がる兆候ととらえ, 欠陥の存在が予想される箇所を推定し, その範囲に対してレビューを実施する[69]. 欠陥の存在リスクが高い部分に対してレビューを行うため効率的なレビューが可能だが, 指摘の導出の方法ではアドホックレビューやシナリオベースレビューを使用しており, 考慮すべき境界条件を整理するプロセスは持たない.

HAZOP [70]や STAMP/STPA [71]といった安全解析手法では, ガイドワードを利用することで, 設計者が意図した振る舞いからのずれや非安全な状態を引き起こす要因を洗い出し, 対策を検討することができる. HAZOP では設計パラメータやユーザの操作に対してガイドワードをあてはめ, 設計意図からのずれを導出する. HAZOP のガイドワードは化学プラントを想定したもので, レビューにおいて考慮すべき境界条件の構造を意識したものではなく, 製品開発の要求仕様レビューにそのままでは当てはめにくい. たとえば製品の利用環境に起因する非安全な状態を分析者が検討するためには,「利用環境」に含まれる境界条件をブレークダウンしたうえで, ガイドワードを適用する必要がある. 利用環境以外に非安全な状態を引き起こす要因があるとすれば, 利用環境以

外の境界条件も想定する必要があり，ガイドワードを活用して設計者の想定外の外的条件を効率的に分析するためには，境界条件の構造に紐付けたガイドワードが必要である．文献[72]や文献[73]ではソフトウェア製品向けに整理したガイドワードを提案しているが，ガイドワードと境界条件の構造との紐づけに関する議論はない．STAMP/STPAではコントロールストラクチャーを作成し要素間の相互作用に着目した分析を行うが，コントロールストラクチャーにおいて要素として洗い出すべき境界条件の構造は規定がない．境界条件を適切に整理するために，利用環境やユーザを含め境界条件の構造を規定したフレームワークが必要である．

ガイドワードをレビューやテストに応用する方法に関する議論として，文献[74]ではガイドワードをレビューに用いることについての言及があるが，具体的な方法は示されていない．文献[75]はガイドワードを用いたテスト項目の抽出手法を提案しているが，レビューへのガイドワードの応用に関する議論はない．ガイドワードを用いて想定外の条件を分析するというアプローチを要求仕様レビューに適用するには，要求仕様レビューに適した構造のガイドワードを構築し，ガイドワードを利用したレビューのプロセスを具体的に定義する必要がある．

5. 従来のレビュー技法の課題と本研究の位置付け

前述の ISO/IEC20246:2017 [22]に掲載されたレビュー技法のうち，特別な手順のないアドホックレビュー以外に共通するアプローチとして，欠陥検出のためのシナリオやチェックリストを事前に用意し，シナリオやチェックリストに基づいて仕様書を読むことで欠陥を検出するアプローチをとる．シナリオやチェックリストを作成する際に境界条件を考慮して作成することはあるが，出来上がったシナリオやチェックリストを利用するだけのレビューアは，製品の境界条件を整理するプロセスを体験せずにレビューを実施することになる．レビューアがシナリオやチェックリストを利用することで境界条件の考慮漏れに気付く場合もあるが，レビューア自身は考慮すべき境界条件の全体を整理していないため，所与の境界条件の範囲内で仕様の指摘を探すような思考のアプローチを取ることになる．このアプローチは，定義された仕様に関する問題点を指摘することに役立つが，境界条件の想定範囲を広げ，想定外の外的条件に起因する問題を見つけることは難しい．新製品の開発や要求定義の初期段階など境界条件が不明瞭な場合，考慮が漏れている境界条件を想定して問題を見つけることが後工程での手戻りの防止

に効果的だが，従来のレビュー技法のアプローチでは，想定外の境界条件を検出できるかどうかはチェックリストやシナリオの内容に依存するため，その効果は限定的である．

一方，テスト分析の方法論では，ユーザに関する視点や利用環境に関する視点など，様々な視点からテストで考慮すべき外的条件を整理するアプローチを取る．これにより仕様書に記載のない外的条件もテストに盛り込むことができ，想定外の外的条件に起因する不具合を検出するテストケースを設計できる．テスト分析における，境界条件を整理して想定外の外的条件を考慮するアプローチと，境界条件を整理する際の視点は，製品の欠陥の検出に役立つという点でレビューでも活用できるものである．本研究ではこのアプローチに基づく新たなレビュー技法を提案する．

想定外の悪条件を検討するには，安全解析手法におけるガイドワードのように，分析者に対して思考のきっかけとなるワードを与え，分析者の考慮範囲を広げるアプローチが有効である．HAZOP では化学プラントのプロセスにおけるパラメータ，STAMP/STPA ではコントロールストラクチャーから導出したコントロールアクションを想定したガイドワードが用意されているが，要求仕様レビューでガイドワードを活用するには，レビューで考慮すべき境界条件の構造と対応づくガイドワードが必要である．本研究では，レビューアが適切に境界条件を整理できるように，要求仕様レビューで考慮すべき境界条件の構造モデルを提案するとともに，レビューアが想定外の悪条件を検討するのに効果的なガイドワードを，境界条件の構造モデルに対応づく形で提案する．要求仕様レビューで考慮すべき境界条件の構造モデルと，その構造モデルに対応づくガイドワードは，従来のレビュー技法・テスト分析手法・安全解析手法には存在しない，本研究の特徴である．

想定外の悪条件を考慮して要求仕様レビューを行うことで，要求仕様書の不備を見つけるだけでなく，要求仕様書の元になった要求の範囲を拡張することに繋がる．要求仕様書の不備を見つけ要求仕様書へ指摘をフィードバックすることを対象とする従来のレビューとは異なり，本研究で提案するレビューでは要求へのフィードバックを導出することが可能となる．図 2-2 に従来の要求分析・レビュー活動やテスト活動と，本研究が提案するレビュー活動の関係性を示す．テスト分析における境界条件のモデリングのアプローチをレビュー活動に応用することで，要求仕様書のみならず，要求仕様書の元になった要求へのフィードバックが可能となる．

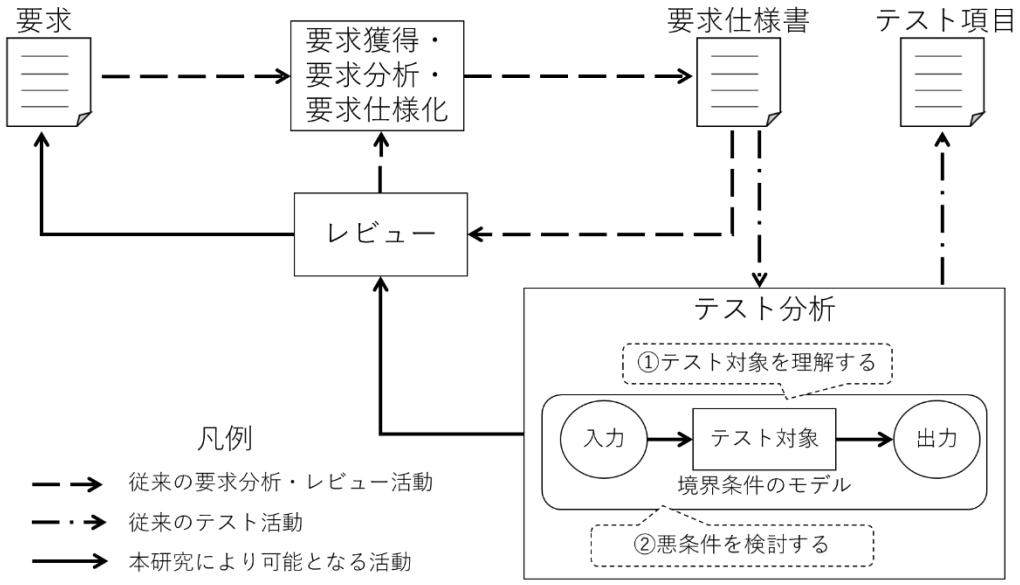


図 2-2　テスト分析を応用したレビュー活動の概念図

6. 結言

本章では，従来のレビュー技法として ISO/IEC20246:2017 [22] や SQuBOK Review2017 [36] に掲載された個人レビューのレビュー技法を紹介した．従来のレビュー技法の多くは欠陥検出のシナリオに基づき仕様書を読んで欠陥を検出するアプローチを取り，シナリオベースドリーディングと呼ばれる手法に属す．このアプローチには，要求仕様で考慮すべき境界条件の整理と悪条件の分析を行うプロセスが明示的でないことを示した．

境界条件の整理を悪条件の検討を行っている検証手法として，ソフトウェアテストのテスト分析および安全性解析の手法を紹介した．テスト分析の手法では，境界条件を整理するフレームワークやツリー構造の記法を利用し，テスト担当者が仕様書の記述からテスト対象の製品の役割・機能・ユーザ・外部環境を理解したうえで，製品の役割・機能の達成を阻害する想定外の悪条件を検討する．テストで考慮すべき境界条件を整理していく過程で仕様書の欠陥を検出することがあり，テストの視点で仕様書を分析することが，一種のレビューの効果を生んでいると考えられる．フレームワークを利用し境界条件を整理するアプローチは要求仕様レビューに応用可能である．安全性解析の手法では，ガイドワードを利用することで，設計者が意図した振る舞いからのずれや非安全な状態を引き起こす要因を洗い出し，対策を検討することができる．ガイドワードを利

用して想定範囲を広げるアプローチは，要求仕様レビューにも応用可能であるが，既往の安全性解析の手法では境界条件の構造とガイドワードの対応付けに関する議論はなく，要求仕様レビューに適した境界条件のフレームワークと，境界条件の構造に対応づくガイドワードの構築が必要である．

第３章　悪条件分析に基づく要求仕様レビュー技法の構築

1. 序言

本章では要求仕様レビューの概念設計を行い，概念設計に基づき，悪条件分析に基づく要求仕様レビュー技法を構築し提案する．概念設計では，テスト分析の思考，特にテスト対象の理解に基づいて悪条件を導出する思考に着目した．要求仕様レビューを「レビューアが境界条件の理解と悪条件の分析を行い，悪条件への対策案を検討して仕様書へフィードバックを行う活動」と捉え，仕様の設計者・レビューア・レビュー対象製品のユーザの関係性を定義し，概念モデルを構築する．さらに，要求仕様レビューで考慮すべき境界条件の概念モデルおよび，悪条件の概念モデルを定義し，境界条件と悪条件の関係性を明らかにする．これらの概念設計に基づき，レビュー技法のプロセスおよび各プロセスにおける入出力となる情報を定義し，レビュー技法として体系化する．各プロセスで利用するフレームワークとして，境界条件を分析するためのフレームワークである Boundary Breakdown Structure，悪条件を導出する際の視点，悪条件への対策案を検討する際の観点を提案する．

2. 要求仕様レビューの概念モデル

概念モデルを設計するにあたり，要求仕様レビューに関連するステークホルダーと，レビューの際に考慮すべき概念の関係性を整理した．ステークホルダーとして，製品を利用するユーザ，要求仕様書を作成する設計者，要求仕様書をレビューするレビューアが存在する．要求仕様書には「要求仕様」が表現される．要求仕様はユーザの「要求」に基づき，設計者によって仕様化される．ユーザの「要求」の背後には，ユーザが製品を利用することで達成したい「ユーザの目的」が存在する．レビューアは要求仕様書を読むことで，要求仕様に表現された「境界条件」を把握する．境界条件がユーザの目的と対応するものであれば，要求仕様はユーザの要求を満たすものとなる．境界条件とユーザの目的の対応関係を阻害するのが「悪条件」である，レビューアは悪条件を想定することで，ユーザの目的が達成されない可能性を見出し，悪条件への対策案を要求や要求仕様へフィードバックする．

これらの概念の関係性を図 3-1 の概念モデルとして整理した．製品開発における要求仕様は，上位の要求を実現するための下位の実現手段ととらえることができる．文献

[76]では人工物を科学として取り扱う方法論が示され，上位のシステムが下位のシステムにより実現される際の対応関係がモデル化されている．本研究では文献[76]による上位のシステムと下位のシステムの対応関係を，要求と要求仕様の関係性として捉え直し，図 3-1 の概念モデルを構築した．「①ユーザの要求」を仕様化することで設計者は「②要求仕様」を作成する．レビューアは要求仕様書を読み境界条件を理解することで「③仕様に制約（境界条件）を与えたモデル」を作成する．レビューアは「③仕様に制約（境界条件）を与えたモデル」を元に悪条件を検討する．悪条件は「③仕様に制約（境界条件）を与えたモデル」と「④要求によって実現されること（ユーザの目的）」の対応関係を崩し，ユーザの目的の達成を阻害する．レビューアは「③」と「④」の対応関係を確立するために，悪条件への対策案を「①」または「②」へフィードバックする．

ユーザは製品開発の専門家でない場合が多く，設計者の領域にある要求仕様を読んでも正確に境界条件を把握することは難しい．設計者にとっては，自らが要求仕様で定義した境界条件に対して，想定外の悪条件を考慮することは難しい．製品開発に精通し設計者とは異なる第三者がレビューアとなることで，ユーザの領域である「④」と設計者の領域である「③」の対応関係を考慮しフィードバックを導出できる．レビューアはユーザと設計者の橋渡しの役割を担う．

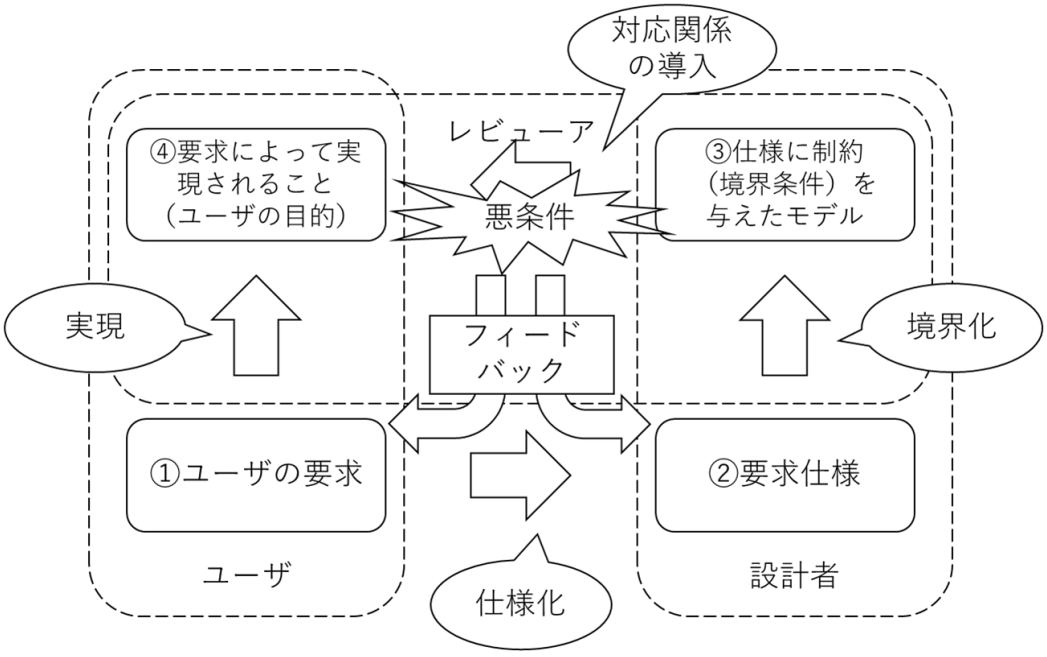


図 3-1　要求仕様レビューの概念モデル

3. 要求仕様レビューで考慮すべき境界条件の整理と Boundary Breakdown Structure の提案

境界条件に関する議論として文献[77]は，システムや制度の設計においては当該の問題状況に関連する様々な境界条件に応じた，複数のアスペクトからモデルを構築する必要があると述べている．システムの境界に関する概念は要求工学の領域でも議論されており，文献[8]では，図 3-2 に示すように，考慮すべき境界条件の外側を環境，環境のうちシステムに影響を及ぼす要素をコンテキスト，環境とコンテキストの境界をコンテキスト境界，コンテキストとシステムの境界をシステム境界と定義している．システムはコンテキストと相互作用を及ぼすための入出力のインタフェースを持つ．要求仕様のレビューを行うには，コンテキスト境界およびシステム境界を形成する境界条件を複数の視点からブレークダウンするための枠組みが必要となる．

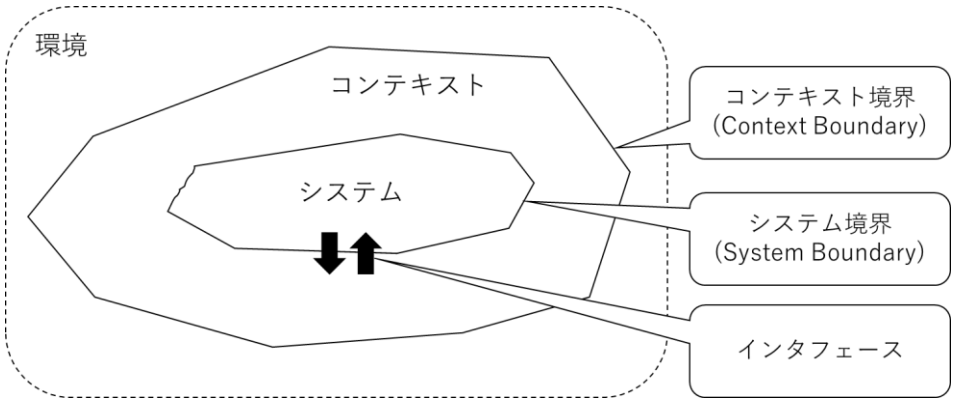


図 3-2　コンテキスト境界とシステム境界
※要求工学知識体系（REBOK）第 1 版[8]の図 1.1 を一部修正して引用

要求仕様で考慮すべき境界条件を，コンテキスト境界およびシステム境界の内部と外部の要素に分け，図 3-3 に示す概念モデルを構築した．文献[78]は組み込みソフトウェア開発における非正常系の要求仕様を分析する手法を提案しており，そのなかで分析対象であるシステムの構成を，機能，ハードウェア構成，環境，運用の 4 つに分けている．図 3-3 でも同様の要素をカバーしているが，(1)環境の要素を外部環境とオブジェクトに分けた点，(2)運用に関連する要素をエージェントの行動モデルとして扱う点，(3)製品ライフサイクルという時間軸による境界条件の変化を概念として取り入れている点が異なる．

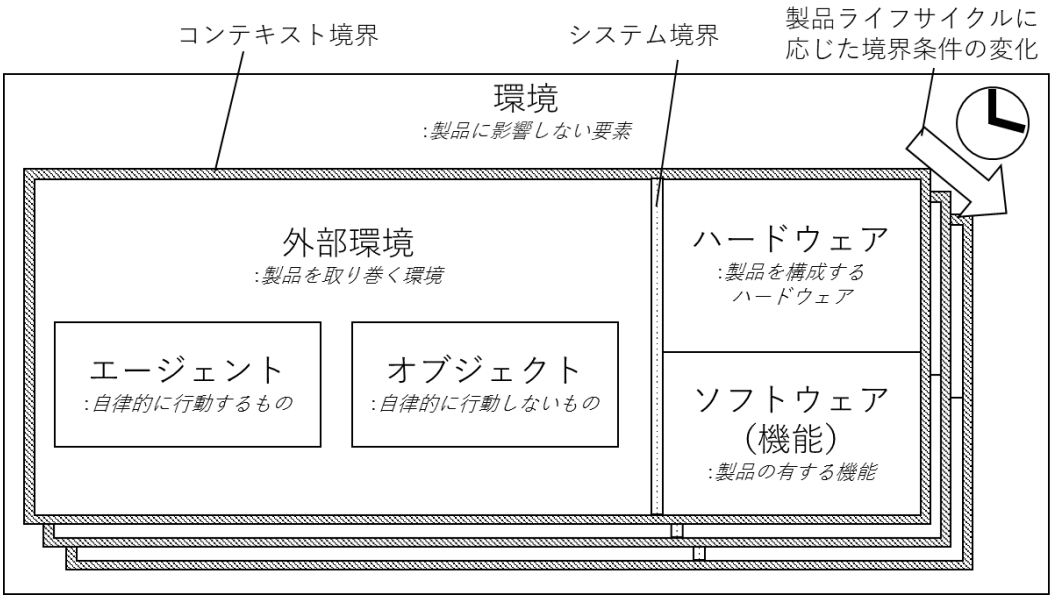


図 3-3　要求仕様で考慮すべき境界条件の概念モデル

(1) に関しては，システムが影響を与えたり影響を受ける物体はオブジェクトとして扱い，温度や湿度などは外部環境として扱うことで，システムに関連する物体そのものと，システムおよび物体の周囲を取り巻く環境を分けて検討でき，検討漏れを防ぐことができる.

(2) に関しては製品の運用者に関わらず，製品に関わりうる主体をエージェントとして表現することで，様々な関係者の行動モデルを表現できる. 製品の運用に関わる主体を表す用語に「ユーザ」があるが，ユーザという用語からは一般的には製品を実際に利用する顧客を想像するため，6W2H における Whom（顧客の顧客）の考慮が漏れやすい. 6W2H の Whom に該当するアスペクトを明示的に示しても良いが，BtoC の製品では顧客の顧客を考慮しなくて良いケースも存在するため，レビューアが概念モデルを利用する際に戸惑うなど，概念モデルの理解性の低下が懸念される.「顧客」も「顧客の顧客」もどちらも製品に関連する主体であるため，両者を含む概念として，自律的に行動する主体を意味する「エージェント」を図 3-3 に採用した. 例えば，パン屋における会計業務を支援するシステムについて考えると，関連するエージェントとして，直接のユーザであるパン屋の店員のほか，パンを購入する客を想定できる. この場合はパン屋の店員が顧客，パンを購入する客が顧客の顧客にあたる. このように，製品の周辺に

存在するエージェントを考慮することで，顧客だけでなく顧客の顧客も考慮可能となる．

(3) に関しては，運搬，設置，運用，廃棄，リサイクルといった製品ライフサイクルのフェーズを考慮することで，フェーズごとに境界条件を検討できる．フェーズが変わると，関連する外部環境，エージェント，オブジェクトが変化し，製品が備えるべき機能も変わる可能性があるため，要求仕様レビューにおいて製品ライフサイクルに応じた境界条件の変化を考慮することは必要である．

具体例として，ボールペンの開発を想定した際に考慮すべき境界条件を，概念モデルにあてはめると図 3-4 となる．ハードウェアとしてはペン先，グリップなどのハードウェア部品が考えられる．ボールペンの機能としてはインクで文字等を書く機能，インクを交換する機能などが備わっている．エージェントとしては，ボールペンを利用する会社員や小学生のほか，幼児など想定外の行動をとるエージェントを想定する必要がある．外部環境としては，温度，気圧，湿度，明るさなどボールペンを使用する際に関連する環境が考えられる．この例から，図 3-3 の構造に沿って境界条件を整理できることが分かる．

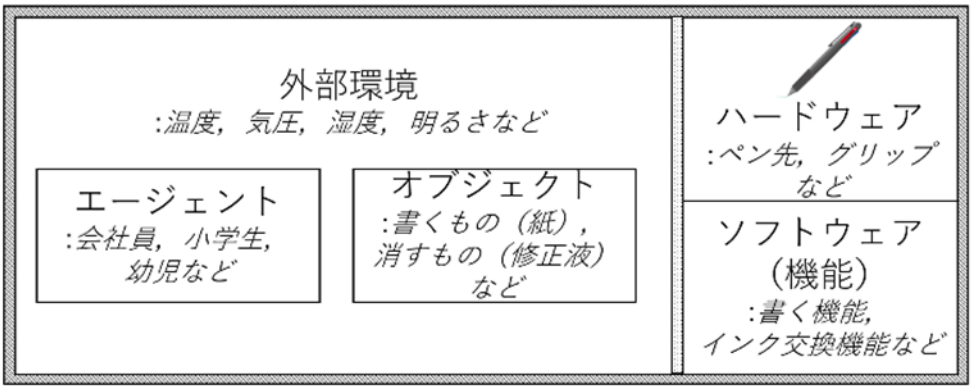


図 3-4　ボールペンの境界条件

本研究では図 3-3 の概念モデルに基づき境界条件を整理するための図法として，Boundary Breakdown Structure を提案する．以降は省略して BBS と記載する．BBS のフレームワークを図 3-5 に示す．図 3-3 で示した境界条件の要素をツリー構造でブレークダウンしながら記載できように，BBS のフレームワークはツリー構造としている．図 3-3 で示した境界条件を明示的にツリー構造で記述するために，「製品ライフサイクル」「エージェント」「システム」「オブジェクト」「外部環境」を第 1 階層のノードとして規定した．

「システム」の第 2 階層として「ハードウェア」と「ソフトウェア（機能）」を規定し，ハードウェアに関してはシステムとコンテキストの間の入出力インタフェースを整理できるように，「入力インタフェース」と「出力インタフェース」を規定した．API のようなソフトウェア的なインタフェースは「ソフトウェア（機能）」の一部としてレビューアが製品ごとに記載することを想定し，規定のノードには含めなかった．レビューアは規定のノードは変更せず，規定のノードの下位の矩形のノードを具体的な値に編集や追加することで，外部環境や利用者等，レビュー対象の境界条件を整理することができる．フレームワークを利用することで，レビューアは規定の構造にそって境界条件をブレークダウンでき，境界条件の考慮漏れを減らすことができる．

製品ライフサイクルのノードでは，ライフサイクルのフェーズに応じたユースケースおよび，ライフサイクル全体を通しての製品のトレーサビリティや製品に関する情報の開示方法を記載する．エージェントのノードでは，製品を利用するユーザなど，製品のステークホルダーを列挙する．システムのノードでは，製品を構成するハードウェアと，ソフトウェアまたは製品が有する機能を要素として抽出する．オブジェクトのノードでは，物理的に製品から影響を受けたり製品に影響を与える物体を検討する．外部環境のノードでは温度や湿度など製品と関連する環境を表現する．

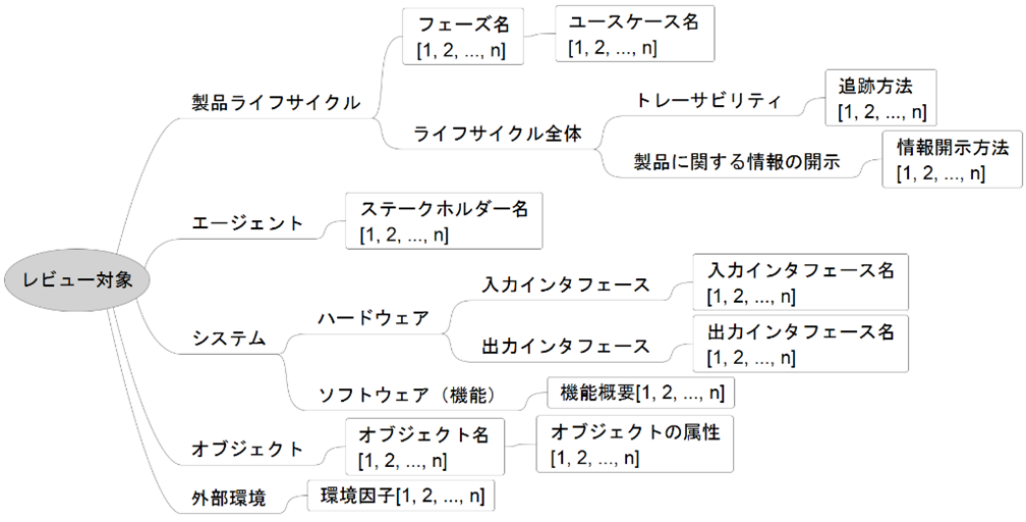


図 3-5　Boundary Breakdown Structure のフレームワーク

BBS の「ユースケース名」のノードをより深く分析するための，ユースケースのフレームワークを図 3-6 に示す．ユースケースは製品ライフサイクルを考慮してエージェントごとに分析する．ユースケースには，レビュー対象を利用する目的，場所，時間，使い方など複数の境界条件が含まれるため，ユースケースのフレームワークではこれらの境界条件を分解して記述できる必要がある．図 3-6 に示したユースケースのフレームワークは 5W1H の構造としており，What は製品，Who はエージェントを表し，当該エージェントのユースケースごとに，製品を利用する目的（Why），利用する場所（Where），利用する時間（When），使い方（How）をブレークダウンして記載できる．ユースケースの記法としては UML のユースケース図やユースケース記述が一般的に知られる[79]が，これらの記法では 5W1H のうち Where，When が明示的に表現されない．製品を利用する場所や時間に応じて想定外の悪条件が発生する可能性があるため，本研究では 5W1H のフレームワークを採用した．ユースケースの分析においては 1 つのエージェントに対して複数のユースケースを記載することが考えられ，BBS にすべてを記載すると BBS が大きくなりすぎて俯瞰性が下がる．そのため，本研究では BBS と別にフレームワークを用意し，ユースケースは別の図として整理することとした．

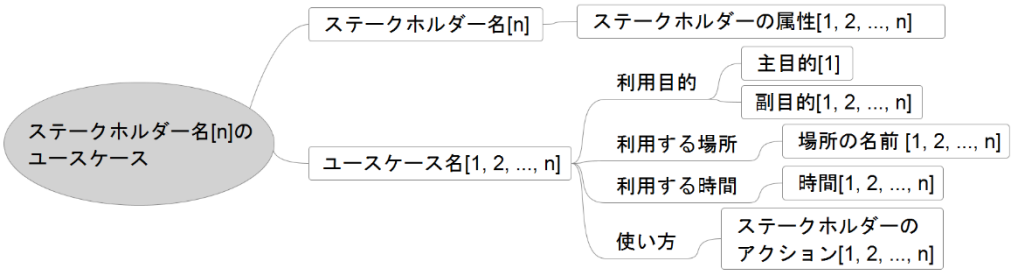


図 3-6　ユースケースのフレームワーク

4. 悪条件の定義と要求仕様へのフィードバックのタイプの整理

悪条件は境界条件として想定された所与の外的条件の範囲を超える条件である．所与の外的条件のもとで設計された製品の仕様は，悪条件が存在することによって満たされなくなる．例えば，Wifi 通信機能を備えた製品を設計する場合，製品を利用する場所としての建物環境を考えると，まずは標準的な環境が境界条件として想定される．それに対し，特別な金属などの電波を遮る物体が存在した場合，標準的な境界条件のみ考慮して製品の仕様が設計されていると，その仕様に基づき開発された製品は正しく

通信することができず，仕様で規定された通信距離を満たさなくなりユーザからのクレームに繋がる可能性がある．設計者の想定外の悪条件を考慮することで，設計者に仕様の問題点に気付かせ，修正を促すことが可能となる．

図 3-7 に悪条件の概念を図式化した．点線の内側が要求仕様書で考慮された境界条件の範囲を表し，点線の外側の網掛けの領域が悪条件の範囲を表す．悪条件はレビュー対象の外部から与えられ，BBS のフレームワークの構成要素のうちソフトウェア（機能）を除く，外部環境・エージェント・オブジェクト・ハードウェアから悪条件が発生しうる．レビュー対象製品がソフトウェアの場合，ハードウェアに関連する条件もソフトウェアにとっての外的条件となる．

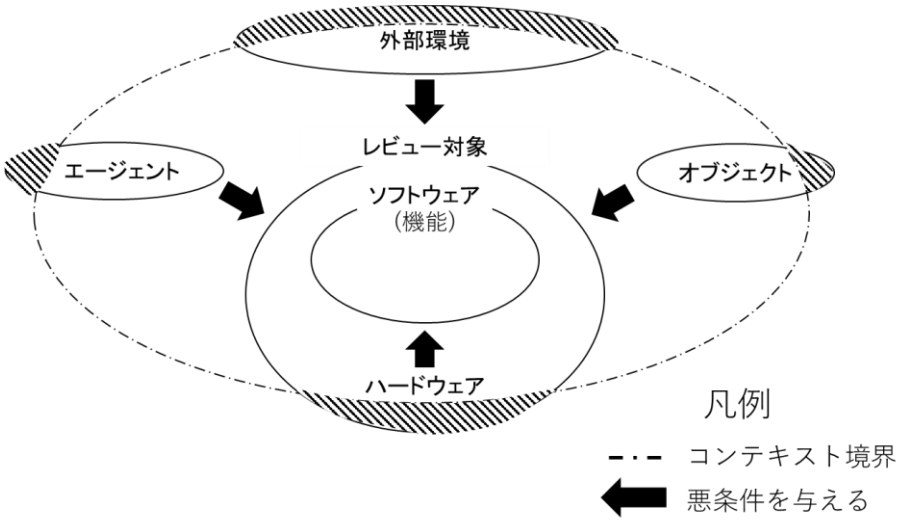


図 3-7　悪条件の概念モデル

製品のリリース後に想定外の問題が発生することを防ぐには，想定外の境界条件をいかにして想定の範囲に納めることができるかが重要である．悪条件を想定して対策を検討し，その対策を仕様として新たに定義することで，レビュー前の要求仕様書で定義された境界条件の範囲，すなわち図 3-7 の点線の内側の領域が拡張され，要求仕様として想定できた範囲が広がる．悪条件を考慮して要求仕様書をレビューすることで，製品リリース後に想定外の問題が発生する可能性を減らすことが可能となる．

製品の仕様を設計する際にどの外的条件まで境界条件として考慮するかは，設計者の知識や経験によって異なることが往々にしてあり，ある外的条件が境界条件なのか

そうでないのかを判断する基準を厳密に定義するのは難しい．仕様のレビューにおいては，レビューアが認識した外的条件が設計者により想定された境界条件なのか想定外の悪条件なのかについて，レビューアが主観的判断で決定する必要がある．境界条件の厳密な判定基準を作成することは難しいが，レビューアが所与の境界条件を認識して悪条件を明示的に検討することで，設計者とレビューアが想定する境界条件の範囲にギャップがないか議論できるようになることが重要である．仕様の設計とレビューの際にこのような議論を行うことによって，次のようなリスクを低減することができる．(1)仕様の設計の後工程で欠陥が見つかり，手戻りによって大きなコストが発生するリスク，(2)市場に製品をリリースしてから想定外の悪条件下で製品が利用されることで欠陥が顕在化し，大きな改修コストが発生するリスク．

レビューアが悪条件を検討する際，製品のユースケースを考慮する必要がある．例えば，ボールペンの製品開発時に考慮される外部環境の悪条件として，真空や超低温といった環境を想定することができる．しかし，製品のユースケースとしてそのような環境で使用されることがなければ対策案を検討する必要がない．このような悪条件は図 3-2 におけるコンテキスト境界の外側を考慮していることになる．悪条件の考慮範囲を広げて製品開発の仕様に境界条件を組み込むほど，製品開発のコストが高くなってしまうため，製品のユースケースを想定して，現実的な範囲で悪条件を検討する必要がある．

ある悪条件に対する対策案は，悪条件を回避する機能を製品に盛り込むこと，悪条件の影響を受けないような外部環境での利用に限定すること，ユーザによる運用でカバーすることなど，様々な観点から検討できる．様々な観点から，要求仕様の改善に繋がる具体的な対策案を検討できるようにするため，レビューアが考慮すべき対策案のタイプを定義する．

何かしらの問題への対策を検討する際，(1)どんな対策を(2)どこに対して施すか，の 2 点の考慮が必要である．「(1)どんな対策を施すか」については，情報セキュリティマネジメントにおけるリスクへの対策のタイプとして，「リスクの低減」「リスクの保有」「リスクの回避」「リスクの移転」の 4 つが知られる[80]，悪条件はユーザの目的の達成を脅かすリスクと考えられるため，前述の 4 つの対策のタイプは悪条件への対策を検討する際にも利用可能と考えた．「(2)どこに対して対策を施すか」については，悪条件の発生源であるエージェント，外部環境，ハードウェア，オブジェクトが考えられる．ハードウェアについては，ソフトウェア（機能）を含む製品の仕様に対して悪条件への対策案をフィードバ

ックすることから，ハードウェアとソフトウェアをあわせた「システム」を対策のポイントとした．外部環境とオブジェクトについては，どちらもシステム外部の要素であることから，これらを合わせて「外部環境」という一つの対策のポイントとした．エージェントについては，製品のユーザへ行動変容を促すことが悪条件への対策の一つになることから，「ユーザ」を一つの対策のポイントとした．

4 つの対策のタイプと，3 つの対策のポイントを考慮して，悪条件への対策案検討の観点を表 3-1 に定義した．対策のタイプと対策のポイントの組み合わせは 12 パターン存在する．「悪条件の低減」では悪条件が発生しても問題が大きくならないような対策案を，ユーザ・システム・外部環境に対して検討可能なため，それぞれに対して対策案検討の観点を挙げた．「悪条件の保有」にでは，悪条件への対策を行わないためシステムおよび外部環境への対策案は存在せず，ユーザに対して悪条件の存在を理解してもらうことのみを対策案検討の観点とした．「悪条件の回避」では悪条件が発生しないようにする対策案を，ユーザ・システム・外部環境に対して検討可能なため，それぞれに対して対策案検討の観点を挙げた．「悪条件の移転」では別の製品や手段でユーザの目的を実現することを対策案として検討するため外部環境への対策案は存在せず，ユーザが別の方法を使うか別のシステムを使うことを対策案検討の観点とした．

表 3-1　悪条件への対策案検討の観点

対策のタイプ	概要	対策のポイント	対策案検討の観点
悪条件の低減	悪条件が発生しても問題が大きくならないように対策する	ユーザ	悪条件が発生した場合の運用手順が明らかになっているか？
		システム	悪条件が発生した場合の対策が取れているか？
		外部環境	悪条件を弱める対策が取れているか？
悪条件の保有	影響がそれほど大きくないため，対策しない	ユーザ	悪条件が存在することをユーザに理解してもらっているか？
		システム	－
		外部環境	－
悪条件の回避	悪条件が発生しないようにする	ユーザ	悪条件を回避するための運用手順が明らかになっているか？
		システム	悪条件に該当する条件をシステムが受け付けないようになっているか？
		外部環境	悪条件が発生しない環境でのみシステムを使用することが決められているか？
悪条件の移転	悪条件の影響を受けない実現手段に変更する	ユーザ	悪条件の影響を受けずに，「達成すべきこと」をユーザが達成する方法はあるか？
		システム	悪条件の影響を受けずに，「達成すべきこと」を達成する別のシステムがあるか？
		外部環境	－

レビューアが悪条件への対策案を検討しやすいように，表 3-1 の悪条件への対策案検討の観点を元にして，対策案のサンプルを表 3-2 として定義した．レビューアが対策案を検討する際，悪条件への対策案のサンプルを元に検討することで，様々な観点で悪条件への対策案を導出できる．4 つの対策のタイプと 3 つの対策のポイントを掛け合わせた構造は表 3-1 も表 3-2 も同様だが，表 3-2 では対策のタイプの概要説明を削除して情報量を減らし，対策案のサンプルを読むだけでレビューアが悪条件への対策案を検討できるようにした．

表 3-2　悪条件への対策案のサンプル

対策のタイプ	対策のポイント	対策案のサンプル
悪条件の回避	ユーザ	悪条件が発生しないように，○○という手順をユーザに通知してはどうか
	システム	悪条件が発生しないように，システムが○○をできるようにしてはどうか
	外部環境	悪条件が発生しないように，外部環境に対して○○という対策を行ってはどうか
悪条件の保有	ユーザ	悪条件への対策がないことを事前にユーザに通知し，理解したうえで使ってもらってはどうか
	システム	-
	外部環境	-
悪条件の低減	ユーザ	悪条件が発生した場合の対策として，○○という手順をユーザに通知してはどうか
	システム	悪条件が発生した場合の対策として，システムが○○をできるようにしてはどうか
	外部環境	悪条件が発生した場合の対策として，外部環境に対して○○という対策を行ってはどうか
悪条件の移転	ユーザ	悪条件の影響を受けないように，ユーザが○○という別の方法を行ってはどうか
	システム	悪条件の影響を受けないように，○○という別の機能を作るか，別の製品・システムを使ってはどうか
	外部環境	-

ボールペンの開発を想定した場合の悪条件への対策案の具体例として，「規定の温度・湿度・気圧の範囲外」という外部環境の悪条件に対して，「悪条件の回避」と「ユーザ」に関する対策案のサンプルを元に対策案を検討することで，「パッケージに規定の温度・湿度・気圧の範囲を記載してユーザに通知する」という対策案を導出できる．「交換用の芯を買う際に正しい芯がどれか分からない」というエージェントの悪条件に対して，「悪条件の低減」と「システム」に関する対策案のサンプルを元に対策案を検討することで，「芯に型番を刻印する」という対策案を導出できる．

5. 悪条件分析に基づく要求仕様レビュー技法のプロセス

本研究で提案するレビュー技法のプロセスの概要を図 3-8 に示す. 従来手法はチェックリストや欠陥検出のシナリオを元に仕様書を読むアプローチを取るのに対し, 提案手法は仕様書の記述を元にレビューアが境界条件を BBS およびユースケースとして整理し直し, 悪条件を検討したうえで要求仕様書への指摘を検討する. 従来のレビュー技法は仕様書の読み方を規定しているためリーディング技法とも呼ばれるが, 提案手法は仕様書を単に読むのではなく, 仕様書に記載されている境界条件を分析して整理し直すアプローチを取ることから,「悪条件分析（Adverse Conditions Analysis）」と呼ぶこととする. 本研究で提案するレビュー技法は「悪条件分析に基づく要求仕様レビュー技法」と呼ぶ. 以降, Adverse Conditions Analysis の頭文字を取り, 提案手法を「ACA」と記載する.

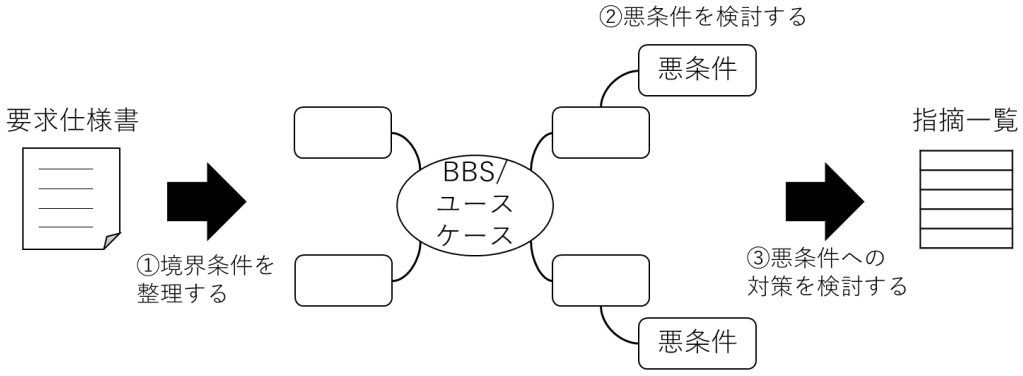


図 3-8　悪条件分析に基づく要求仕様レビュー技法のプロセスの概要

提案手法によるレビューのプロセスを図 3-9 に示す. まず要求仕様書をインプットとし, レビュー対象の境界条件を識別するために BBS を作成する. BBS を作成する際, 図 3-5 のフレームワークに沿って境界条件を整理することで, レビュー対象製品を構成するハードウェアやソフトウェアや機能のみならず, エージェント, オブジェクト, 外部環境といった, 製品に関連する境界条件の全体を識別できる. フレームワークで規定された「製品ライフサイクル」「エージェント」「システム」「オブジェクト」「外部環境」の各ノードは変更せず, 規定のノードの下位にある矩形のノードに具体的な値を代入したりノードを追加することで, 要求仕様で考慮すべき境界条件を分析する. BBS は 1 個のレビュー対象に対して通常は 1 個のツリー構造の図を作成する. レビュー対象の機能が多い場

合など BBS が巨大になり過ぎる場合は，適宜，複数の図に分割して記載することができる．

BBS に記載したエージェントごとにユースケースを特定する．ユースケースを特定する際，図 3-6 のフレームワークに沿って分析する．フレームワークで規定された「利用目的」「利用する場所」「利用する時間」「使い方」の各ノードは変更せず，その他のノードに具体的な値を代入したりノードを追加することで，ユースケースを分析する．1 つのエージェントに対して複数のユースケースが存在する場合があり，BBS が大きくなりすぎて俯瞰性が下がるのを防ぐため，BBS とは別の図としてユースケースを表現している．ユースケースは 1 個のエージェントに対して通常は 1 個のツリー構造の図を作成し，複数のユースケースがある場合はノードを分けて記載する．1 個のエージェントに対するユースケースが多く，ユースケースが巨大になり過ぎる場合は，1 個のエージェントに対して図を複数個作成することができる．

BBS とユースケースから悪条件を検討して一覧化する．悪条件を検討する際は，図 3-7 の概念モデルに示したように，ハードウェア，オブジェクト，外部環境，エージェントのアスペクトから，各境界条件と関連する想定外の外的条件を検討する．悪条件の導出のプロセスは BBS およびユースケースの作成の後に位置するが，実際にレビューの作業を実施する際は BBS やユースケースの作成時に悪条件を思いつくことや，部分的に悪条件の導出を行ったあとで BBS やユースケースをブラッシュアップすることもありえるため，これらのプロセスは行き来することができる．悪条件を思いついた時点で，BBS やユースケースのノードの末端に文字色やフォントを変更して悪条件のノードを作成しておくと，効率的に悪条件の導出を進めることができる．悪条件を導出する際に，安全性解析の手法のようにガイドワードを用いることでレビューアが悪条件を導出しやすくなる．悪条件導出のガイドワードは次章にて構築する．

最後に，個々の悪条件に対して，その悪条件が発生した際に現状の要求仕様で問題が発生するか否かをレビューアが判断し，問題となる悪条件への対策案を検討し，要求仕様への指摘として一覧化する．悪条件への対策案を検討する際は，表 3-2 の対策案のサンプルを参考にすることで，様々な対策のタイプと対策のポイントを考慮して対策案を導出できる．

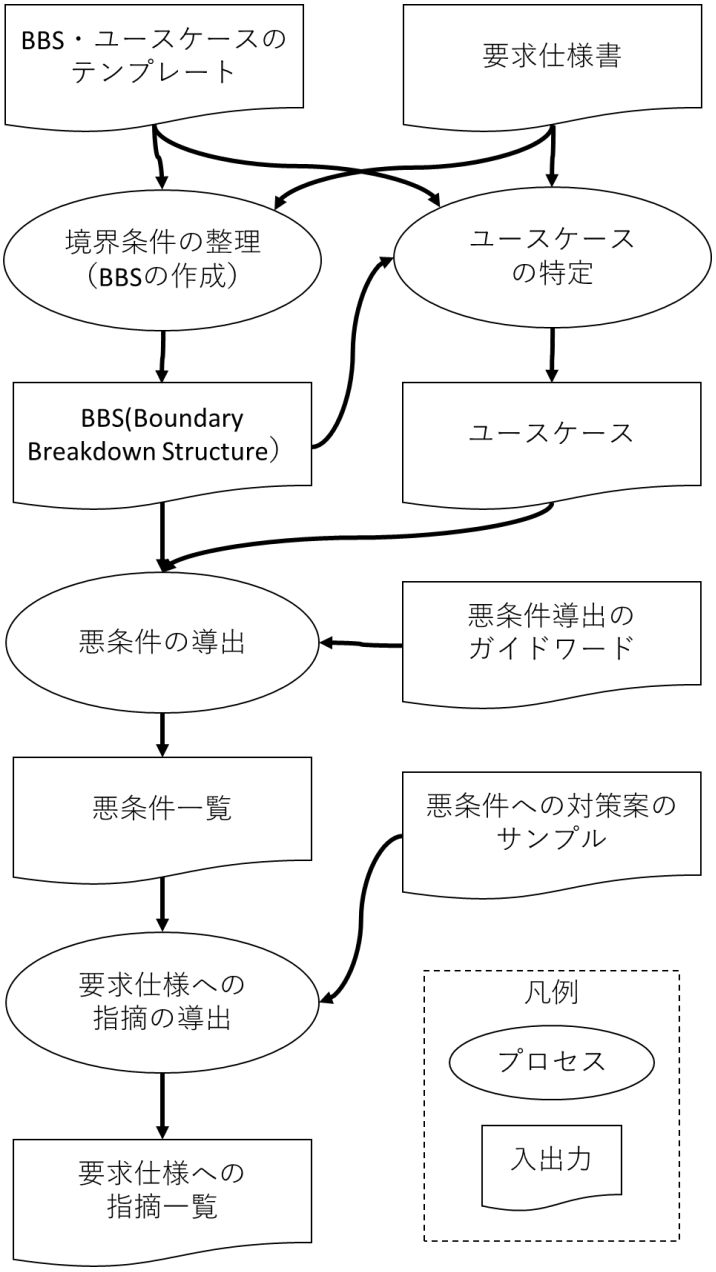


図 3-9　悪条件分析に基づく要求仕様レビュー技法のプロセス

6. 結言

本章では，要求仕様レビューの概念モデルを定義するとともに，要求仕様レビューで考慮すべき境界条件の概念モデル，悪条件の概念モデル，悪条件への対策案を検討する観点を提案した．境界条件の構成要素としてハードウェア，ソフトウェア（機能），エージェント，オブジェクト，外部環境，製品ライフサイクルが存在することを概念モデル

により示し，これらの要素をブレークダウンしながら階層構造で分析するためのフレームワークとして，Boundary Breakdown Structure（BBS）を提案した．ユースケースを分析するためのフレームワークとして，5W1H の構造を活用したフレームワークを提案した．悪条件の概念モデルでは，ハードウェア，オブジェクト，エージェント，外部環境の各要素から，ソフトウェア（機能）に対して悪条件が与えられるモデルを設計した．この概念モデルにより，悪条件はハードウェア，オブジェクト，エージェント，外部環境の 4 つの視点から導出できることを示した．悪条件への対策案検討の観点は，リスク対策の 4 つのタイプである低減，保有，回避，移転と，対策のポイントであるユーザ，システム，外部環境の 3 つを掛け合わせることで，多様な観点を定義した．レビューアが悪条件への対策案を検討しやすいように，対策案検討の観点を具体化し，対策案のサンプルを構築した．

これらの概念モデルに基づき，悪条件分析に基づく要求仕様レビュー技法のプロセスを定義した．提案手法は単に仕様書を読んで指摘を導出するのではなく，まず仕様を理解して境界条件を整理することから作業を開始する．境界条件の整理の結果に基づき，境界条件の範囲を超える悪条件を想定し，悪条件への対策案を要求仕様書への指摘として導出する．境界条件の整理と悪条件の検討という分析的な思考を用いることで，要求仕様書の設計者が考慮していなかった想定外の悪条件を，レビューアが導出できるようになる．BBS およびユースケースのフレームワーク，悪条件への対策案のサンプルを利用することで，レビューアは多様な観点から要求仕様への指摘を導出できる．

第４章　ケーススタディに基づく悪条件導出のガイドワードの構築

1. 序言

本章では，前章で提案した「悪条件分析に基づく要求仕様レビュー技法」を，2 つの具体的な事例に適用した結果を報告する．事例適用の対象として，ハードウェアのみから構成される製品であるボールペンと，ソフトウェアとハードウェアの両者から構成される製品であるパン自動会計システムを対象とし，2 つの異なるタイプの製品に提案手法を適用可能なことを示す．事例適用により導出した具体的な悪条件を分類・抽象化することで，悪条件導出のガイドワードを提案する．ガイドワードを利用することで，レビューアが当初想定できなかった新たな悪条件を導出できることを示す．

ケーススタディの対象の選択基準として，(1) 筆者がケーススタディを実施するにあたり製品に対する専門的な知識がなくとも境界条件やユースケースを想定でき悪条件を導出できること，(2) ソフトウェアが組み込まれているか否かで悪条件の発生源が異なるため，その両者に該当する製品を対象とすること，の 2 点を基準とした．ボールペンは多くのユーザが日常的に利用していることと，パン自動会計システムは主なユースケースがパン屋でパンを購入する際の会計の手続きに限られることから，いずれの製品も専門的な知識がなくとも境界条件とユースケースを想定でき，悪条件を導出できる．ボールペンはハードウェア部品のみで構成されハードウェアが製品の機能を実現するのに対し，パン自動会計システムはソフトウェアが組み込まれておりハードウェアだけでなくソフトウェアにより製品の機能が実現される．ボールペンのようなハードウェアのみで構成される製品では，ハードウェアは製品そのもののため，外的な悪条件の発生源とならない．一方，パン自動会計システムのようにソフトウェアが製品の機能を実現する場合，ハードウェアはソフトウェアにとっての外的条件になり，悪条件の発生源となりうる．これらの 2 つの製品をケーススタディの対象とすることで，タイプの異なる 2 つの製品に提案手法を適用可能なことを示すことができる．

提案手法を用いて要求仕様書への指摘を導出するには，レビューアが悪条件をいかに想定できるかが重要である．前章で述べたとおり，悪条件はハードウェア，オブジェクト，外部環境，エージェントの 4 つの視点から導出できるが，それらの視点から実際

に悪条件を検討するのは個々のレビューアであり，どれだけ悪条件を想定できるかはレビューアの思考力に依存する．提案手法の効果を十分に発揮するには，広い視点で悪条件を想定できるように個々のレビューアの思考範囲を広げるアプローチが必要である．HAZOP [70]や STAMP/STPA [71]に代表される安全解析手法では，想定外のリスクを洗い出すために，分析者の思考を誘導するキーワードとしてガイドワードを活用する．提案手法においても，悪条件を導出するためのガイドワードを用意することで，有効な悪条件の分析を実施できると考えられる．本章では，ケーススタディで導出した具体的な悪条件を分類・抽象化し，悪条件導出のガイドワードを構築する．

2. ボールペンの要求仕様書を対象としたケーススタディ

ボールペンの要求仕様書をレビュー対象とし，ケーススタディを行った．インプットとした要求仕様書は，ボールペンを開発するにあたっての要求事項を自然言語で記載した文書で，分量は A4 サイズの用紙に印刷すると約 1 ページである．目次構成は，機能に関する要求，使用期間に関する要求，使い勝手に関する要求，使用環境に関する要求，物理的な耐性に関する要求，安全性に関する要求，環境対策に関する要求，から成る．機能に関する要求の一部を以下に示す．社名や品番は仮称である．

(1) 4 色書くことができる（黒・赤・青・緑）
(2) ペン先の向きが水平よりも下向きであれば書くことができる
(3) 交換用の芯の規格は，ZB 社の品番 sk-01 に準拠する
(4) クリップ部は開閉可能とし，ばねを利用して自然と閉じるようにする（クリップ部が折れるのを防ぐため）

ボールペンの要求仕様書をインプットにして作成した BBS を図 4-1 に示す．「システム」の下位ノードは要求仕様書に記載のあるハードウェアおよびボールペンの持つ機能を抽出した．今回対象としたボールペンはソフトウェアを含まないため，「ソフトウェア（機能）」のノードはボールペンの機能と捉えてブレークダウンした．「製品ライフサイクル」のノードは，製品ライフサイクルマネジメントの分野で提唱されているフェーズ[81, 82]を参考とし，製品企画・開発，広告・宣伝，製造，販売・流通，使用，廃棄で分類した．そのうえで，フェーズごとにユースケースを検討した．「エージェント」の下位ノードは，製品ライフサイクルの「使用」に関わるエージェントをメインに検討した．ボールペンの一般的なユーザの他に，オフィス内で働く会社員，屋外で作業するビル工事の現場主任，

極端な環境下で活動する宇宙飛行士，大人とは異なる年齢層として小学生と幼児を導出した．製品ライフサイクルの「販売・流通」に関わるエージェントとして，運送業者や販売店の店員も考えられるが，今回のケーススタディではボールペンを購入して利用するユーザに絞って検討した．エージェントをどこまで考慮すべきかは，レビューのスコープに関連するため，レビューアおよび製品開発プロジェクトのメンバーなどステークホルダーの間で共有し合意する必要がある．「オブジェクト」の下位ノードはユースケースを想定し，レビュー対象以外に存在する物体を検討した．ボールペンを利用する際，ノートや紙といった書きつけるものと，一度書いた文字等を修正するための消すものが存在しうるため，これらをオブジェクトとして導出した．「外部環境」の下位ノードは，要求仕様書の記載から，温度，気圧，湿度を抽出し，製品を利用可能な範囲を記載した．そのほかにユースケースを考慮して，明るさという外部環境条件を導出した．

図 4-1　ボールペンの BBS

ユースケースの事例として，BBS で抽出した「会社員」というエージェントのユースケースを図 4-2 に示す．ユースケースを作成する際は図 3-6 のフレームワークを利用し，フレームワークに規定された「利用目的」「利用する場所」「利用する時間」「使い方」というノードは編集せず，その他のノードに具体的な値を代入したりノードを追加することでユースケースを作成した．

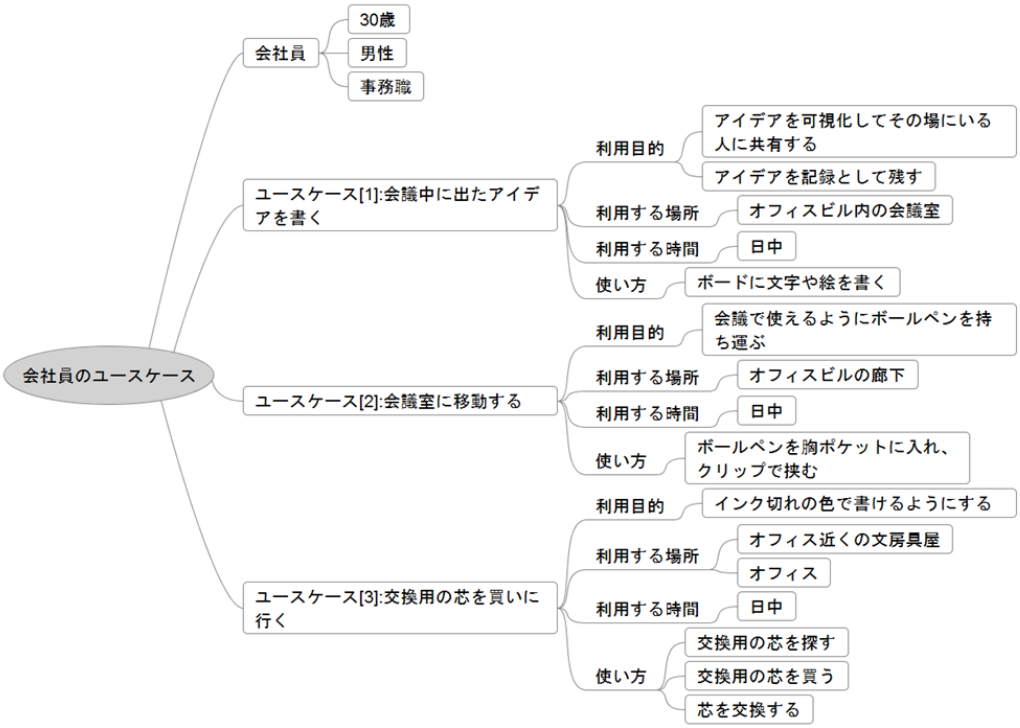


図 4-2　会社員エージェントによるボールペンのユースケース

会社員エージェントの属性として，30 歳の男性であり事務職に就く会社員を想定した．製品開発において設計を支援するために定義する仮想のユーザ像はペルソナと呼ばれる．ペルソナは実際のユーザの調査に基づいて定義されるが，ユーザの調査に基づかずに定義される仮のペルソナは特にプラグマティック・ペルソナと呼ばれる[83]．エージェントの属性を定義することで，そのエージェントはプラグマティック・ペルソナに近づき，レビューアが具体的なユースケースを想定できるようになる．製品開発プロジェクトでペルソナが定義されている場合，ペルソナをエージェントと捉えてユースケースを検討できる．

同一のエージェントに対して複数のユースケースが考えられる場合は，フレームワークの「ユースケース名」以下のノードを増やして，複数のユースケースを記載する．本節のケーススタディでは「会議中に出たアイデアを書く」「会議室に移動する」「交換用の芯を買いに行く」という 3 つのユースケースを検討した．3 つのユースケースそれぞれに対して，製品を利用する目的（主目的・副目的），利用する場所，利用する時間，具体的な使い方を記載した．会社員以外のエージェントとして一般ユーザ，ビル工事の主任，宇宙飛行士，小学生，幼児が BBS で導出されているため，これらのエージェントについても同様の構造でユースケースをそれぞれ作成する．

BBS のノードから検討した悪条件を表 4-1 に示す．各エージェントのユースケースから検討した悪条件を表 4-2 に示す．ボールペンはハードウェアのみから構成され，ハードウェアが製品の機能を担うため，「システム-ハードウェア」の下位ノードに該当する要素は製品内部の要素と考えられ，悪条件の発生源とならない．BBS のノードから悪条件を検討する際は，所与の外的条件に該当する「オブジェクト」と「外部環境」のそれぞれの下位ノードから悪条件を検討した．

表 4-1　BBS のノードから検討した悪条件（ボールペン）

BBS のノード			悪条件
オブジェクト	書くもの	材質	表面がつるつるした紙
			表面がざらざらした紙
			油性インクが乗らない素材
	消すもの	消す方法	修正の上からまた書く
			溶剤で溶かす
	クリップで挟むもの	厚み	規定の厚さの範囲外
		材質	表面がつるつるした物体
外部環境	温度		規定の温度の範囲外
	気圧		規定の気圧の範囲外
	湿度		規定の湿度の範囲外
	明るさ		色の区別がつきにくい明るさ

表 4-2　ユースケースから検討した悪条件（ボールペン）

エージェント	ユースケース	悪条件
会社員	会議中に出たアイデアを書く	水平よりもペン先を上にして書く
	会議室に移動する	ペン先が出たままスーツやワイシャツをクリップで挟む
	交換用の芯を買いに行く	間違った芯を買う
		正しい芯がどれなのか分からない
		交換用の芯が店頭に無い
	オフィスのごみ箱に捨てる	分別せずに捨てる
宇宙飛行士	実験結果を書き留める	無重力下で使用する
		気圧の低い場所で使用する
ビル工事の主任	点検結果を書き留める	雨の中で使用する
		振動が激しい場所で使用する
		土埃が多い場所で使用する
小学生	年賀状を書く	印刷された絵に重ねて文字を書く
幼児	ボールペンに興味を持つ	なめる
		飲み込む
		分解する
		叩く
		意図しない方向に力を加える
		投げる

オブジェクトの悪条件としては，ノートや紙といった文字を書きつけるものの材質として「表面がつるつるした紙」「表面がざらざらした紙」「油性インクが乗らない素材」を導出した．これらは，文字や絵を書くというボールペンの機能を阻害する要因になりうるため，悪条件であると考えられる．外部環境の悪条件としては，仕様で規定された環境の範囲外で利用されるケースを想定し，規定の温度・気圧・湿度の範囲外という悪条件を導出した．「色の区別がつきにくい明るさ」という悪条件は，複数の色を区別して書くという 4 色ボールペンの機能を阻害する要因になりうるため，悪条件として導出した．

3. パン自動会計システムの要求仕様書を対象としたケーススタディ

パン自動会計システムの要求仕様書をレビュー対象とし，ケーススタディを行った．このシステムは，パン屋での会計業務を効率化するために，パンの種別を画像判定によって自動識別し，料金をレジ機に自動入力するシステムである．インプットとなる要求仕様書は，パン自動会計システムへの要求事項を自然言語および図で記載した文書である．要求仕様書の自然言語の分量は，A4 サイズの用紙に印刷すると約 2 ページである．そのほかに画面イメージ 15 枚から構成されている．目次構成は，本文書の位置づけ，システム導入の目的，システムの概要（全体イメージ，ハードウェア，OS，インタフェース），機能要件，画面イメージ，から成る．機能要件の一部を以下に示す．

(1) パンの商品名と単価を登録する

(2) カメラでパンを撮影しパンの種類を識別する

(3) パンを正しく識別できない場合，ユーザがパンの種類を指定する

(4) パン以外の物体がトレイに乗っている場合，会計の対象から除外できる

パン自動会計システムの要求仕様書をインプットにして作成した BBS を図 4-3 に，BBS で洗い出したエージェントのうち店員のユースケースを図 4-4 に，パンを購入する客のユースケースの一部を図 4-5 に示す．ハードウェア部品だけで構成されるボールペンと異なり，パン自動会計システムはハードウェアとソフトウェアにより機能が実現される．悪条件を検討する際，BBS のハードウェア・オブジェクト・外部環境の各ノードおよび，エージェントのユースケースを悪条件の発生源とした．BBS のノードから検討した悪条件を表 4-3 に，ユースケースから検討した悪条件を表 4-4 に示す．

図 4-3　パン自動会計システムの BBS

表 4-3　BBS のノードから検討した悪条件（パン自動会計システム）

BBS のノード			悪条件
ハードウェア	入力インタフェース	カメラ	カメラがフリーズする，カメラが故障する，カメラの前の障害物に邪魔される，カメラのピントが合わず画像がぼやける，本体との接続が切断される
		操作用タッチパネル	タッチパネルがフリーズする，タッチパネルが故障する，タッチしても認識されない，指以外のものに反応する，本体との接続が切断される
	出力インタフェース	ディスプレイ	ディスプレイがフリーズする，ディスプレイが故障する，画面表示が遅い，長時間電源を入れっぱなし，本体との接続が切断される
オブジェクト	トレイ		トレイの色が規定の範囲外，トレイを 2 つ以上重ねる，トレイが汚れている
	パン	種類	種類が規定の数より多い，すでに販売していない種類，新たに販売を予定している種類，試験的に短い期間だけ販売する種類，季節限定商品
		値段	値段が規定の範囲外，値引き
		大きさ	大きいパン（フランスパンなど），小さいパン（ミニドーナツなど）
		形	同じ種類でも形状が異なるパン，潰れている，一部が破損している
		焼き加減	同じ種類でも焼き加減の異なるパン
		パッケージ	ビニールに包まれている，紙やアルミの皿に乗っている
		個数	大量のパン，パンがない
	OS		OS アップデート，OS サポート期限切れ
	何らかの物体		カメラの前に障害物がある，パン以外の物体（トングなど）
外部環境	明るさ		明るすぎる・暗すぎる，場所や時間による明るさの変化

ハードウェアについては，入力インタフェースとしてユーザが操作するための「タッチパネル」とパンを撮影する「カメラ」，出力インタフェースとしてユーザが見るための「ディスプレイ」が存在する．各ハードウェアがフリーズしたり故障したり本体との接続が切断された場合，システムが正常に動作しないため，「フリーズする」・「故障する」・「本体との接続が切断される」を悪条件として導出した．各ハードウェアに固有の悪条件として，カメラは何らかの障害物によって撮影を邪魔された場合や，ピントが合わずパンを適切に撮影できなかった場合に，パンを識別するという機能が達成できないため，「カメラの前の障害物に邪魔される」「カメラのピントが合わず画像がぼやける」を悪条件として導出した．タッチパネルはタッチが正しく認識されない場合にパン自動会計システムの各機能を達成できないため，「タッチしても認識されない」・「指以外のものに反応する」を悪条件として導出した．ディスプレイは情報が適切なタイミングで表示されない場合に機能を達成できないため，「画面表示が遅い」という悪条件を導出した．またディスプレイは長時間電源を入れっぱなしにすることで焼き付きや発熱といった問題に繋がる可能性が考えられるため，「長時間電源を入れっ放し」という悪条件を導出した．

オブジェクトについては，システムで識別する対象である「パン」，パンを乗せる「トレイ」，パンを掴む「トング」などの物体がシステムの周辺に存在する．パンの属性として種類・値段・大きさ・形・焼き加減などが考えられる．パンの種類については，パン屋では多様な種類のパンが存在することから，種類が仕様で規定されている数を超えて業務に支障が出る可能性や，過去に販売していたがすでに販売していない商品や期間限定の商品などのデータをどのように扱うべきかといった問題が考えられ，これらを悪条件として導出した．パンの識別を誤らせる可能性のある，大きいパンや小さいパン，潰れていたり一部が破損しているなど形状が通常と異なるパン，焼き加減の異なるパンなどを悪条件として導出した．

外部環境については，パンをカメラで撮影する際に明るさが影響すると考えられるため，BBS のノードとして「明るさ」が導出された．明るさが適切でないと正しい撮影結果が得られない可能性があることから「明るすぎる・暗すぎる」という悪条件を導出した．明るさは場所や時間によって変化しうるため，ある地点のある時点で適切に撮影できたとしても，別の場所や時間では適切に撮影できない可能性があることから，「場所や時間による明るさの変化」という悪条件を導出した．

エージェントは大きく分けると，会計を行うパン屋の店員とパンを購入する客が考えら

れる．客には様々なタイプが存在し客ごとにユースケースと悪条件が異なることが予想されたため，年齢層や利用状況の異なる複数のエージェントを想定して BBS のノードとした．図 4-4 に示した店員のユースケースでは，パンの識別に必要な画像情報や価格情報を事前に登録するというユースケース，実際に会計を行うユースケースが想定された．パンの写真と価格をシステムに登録するユースケースでは，何も情報が登録されない状態ではパンを識別する機能が達成できないため，「何も登録しない」を悪条件として導出した．また情報を大量に登録する必要がある場合，処理性能の劣化や登録可能な上限を超える可能性があることから，「大量に登録する」を悪条件とした．店員が会計するというユースケースでは，会計がうまくいかない場合を想定して，「トレイを斜めにセットする」「何度も撮影する」を悪条件として導出した．またパンの認識が誤っていた場合の対応を店員が間違える場合を想定して，「誤認識に気付かず修正しない」「間違えて再度修正する」を悪条件として導出した．

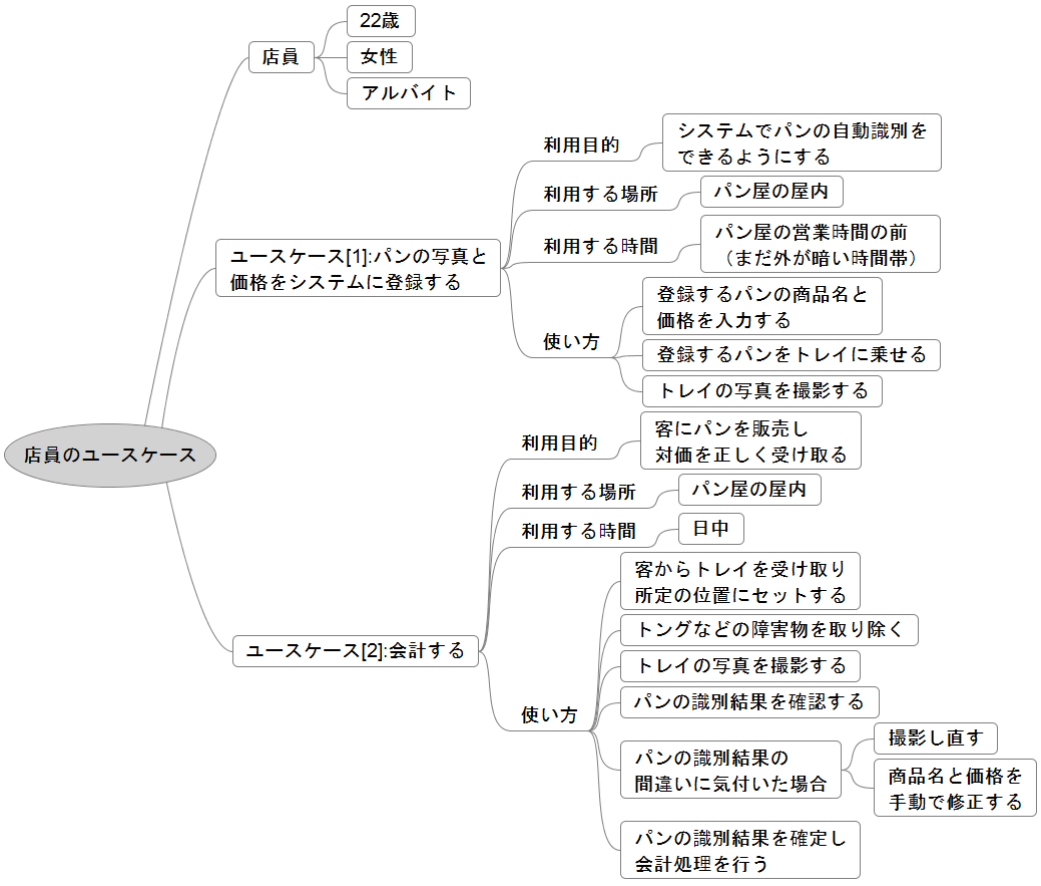


図 4-4　店員エージェントによるパン自動会計システムのユースケース

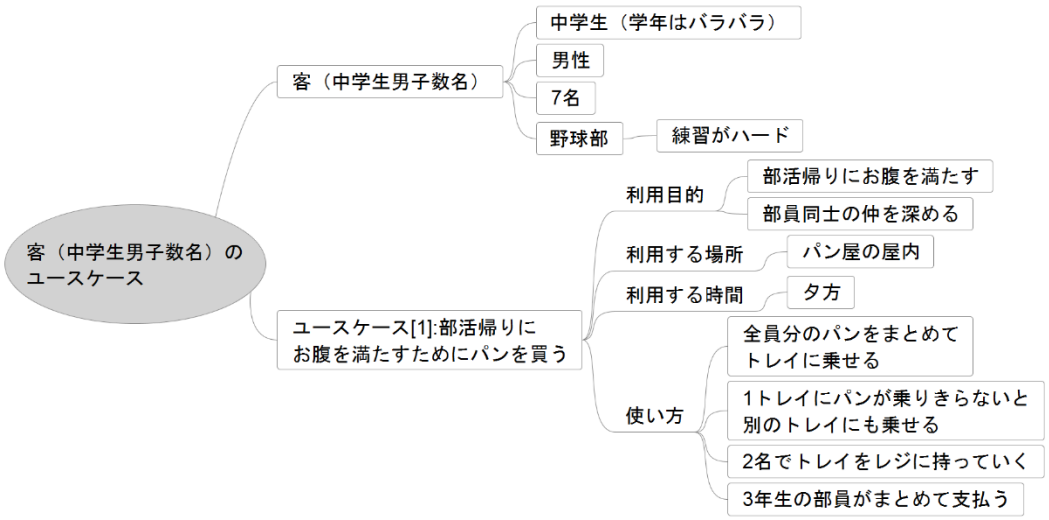


図 4-5　客エージェントによるパン自動会計システムのユースケース

パンを購入する客の悪条件は，各エージェントのユースケースに応じて発生しうる，システムの機能の達成を阻害することに繋がる外的条件を検討した．図 4-5 に示した，中学生男子数名の「部活帰りにお腹を満たすためにパンを買う」というユースケースでは，お腹を満たすという主目的の他に，部員同士の仲を深めるという副目的を想定し，中学 3 年生の先輩が 2 年生や 1 年生の後輩にパンをおごるケースを想定した．この場合，トレイに大量のパンを乗せることになるため，1 つのトレイにパンが乗りきらずに 2 つのトレイを利用することや，パンを重ねて乗せることが考えられるため，悪条件として「2トレイ以上まとめて会計する」と「トレイに重なるほどパンを乗せる」を導出した．そのほかのエージェントについてもユースケースを作成して悪条件を検討した．独身の男性会社員の「自分の昼食用にパンを買う」というユースケースでは，昼食用に袋詰めのサンドイッチと飲み物を買うことを想定し，「サンドイッチなど，袋詰めされた商品をトレイに乗せずにそのまま持ってくる」と「飲み物など，パン以外の商品と一緒に会計する」という悪条件を導出した．子連れの主婦の「子供のおやつと翌日の朝食用にパンを買う」というユースケースでは，おやつ用と朝食用で異なるパンを買うこと，家族それぞれの好みに合わせて多様なパンを買うことを想定し，「色々な種類のパンをいっしょくたに買う」と「フランスパンなど大きいパンを買う」という悪条件を導出した．子供の「お母さんに連れられて来店する」というユースケースでは，会計処理の途中で別のパンを欲しがることを想定し，「会計途中でパンを持ってくる」という悪条件を導出した．子供がトレイを持ちたが

って持たせたが，うまく持てずに落としてしまうことを想定し，「トレイを落として破損する」を悪条件として導出した．既婚の女性会社員が「夜に安売りのパンを買うために来店する」というユースケースでは，夜まで売れ残って割引価格で販売されているパンを家族の分も含めてたくさん買うことを想定し，「同じ種類のパンをたくさん買う」という悪条件を導出した．

表 4-4　ユースケースから検討した悪条件（パン自動会計システム）

エージェント	ユースケース	悪条件
店員	パンの写真と価格をシステムに登録する	何も登録しない
		大量に登録する
	会計する	トレイを斜めにセットする
		何度も撮影する
		パンの誤認識に気付かず修正しない
		パンの誤認識を修正するが，間違えてさらに修正する
客（独身男性会社員）	自分の昼食用のパンを買う	サンドイッチなど，袋詰めされた商品をトレイに乗せずにそのまま持ってくる
		飲み物など，パン以外の商品と一緒に会計する
客（子連れ主婦）	子供のおやつと翌日の朝食用にパンを買う	色々な種類のパンをいっしょくたに買う
		フランスパンなど大きいパンを買う
客（子供）	お母さんに連れられて来店する	会計途中でパンを持ってくる
		トレイを落として破損する
客（中学生男子数名）	部活帰りにお腹を満たすためにパンを買う	2トレイ以上まとめて会計する
		トレイに重なるほどパンを乗せる
客（既婚女性会社員）	夜に安売りのパンを買うために来店する	パンを割引価格で買う
		同じ種類のパンをたくさん買う

4. 悪条件導出のガイドワードの構築

ケーススタディから得られた具体的な悪条件を分類・抽象化し，レビューアが悪条件を導出する際に利用可能なガイドワードを構築する．元来，ガイドワードは HAZOP や STAMP/STPA などのハザード分析の手法で利用され，設計意図からのずれや非安全な状態を引き起こす要因を検討するためのヒントになる語を指す．文献[70]では HAZOP で利用するガイドワードとして，「無（no）」，「逆（reverse）」，「他（other than）」，「大（more）」，「小（less）」「類(as well as)」，「部(part of)」，「早(early)」，「遅(late)」，「前(before)」，「後(after)」の 11 語が示されている．文献[84]では STAMP/STPA で利用するガイドワードとして，「与えられないとハザード（Not providing causes hazard）」，「与えられるとハザード（Providing causes hazard）」，「早すぎ，遅すぎ，誤順序でハザード（Too early, too late, out of order）」，「早すぎる停止，長すぎる適用でハザード（Stopped too soon, applied too long）」の 4 つが示されている．HAZOP ではアクシデントは構成要素の故障やユーザのオペレーションミスが原因で発生すると考え，設計パラメータや正常な振る舞いに対してガイドワードを適用することで，設計意図から逸脱した状態や異常な振る舞いを導出する．しかし，製品の機能の実現手段がハードウェア主体からソフトウェアを含むようになり複雑化したシステムにおいては，個々の構成要素が故障していなくてもアクシデントが発生するケースがある．STAMP/STPA ではアクシデントは構成要素間の相互作用の不備に起因して発生すると考え，アクシデントに繋がる可能性のある非安全なコントロールアクションを導出するためにガイドワードを利用する．提案手法における悪条件は，製品の機能と，エージェント・オブジェクト・外部環境といった製品の外部に存在する境界条件の相互作用において発生すると考えられることから，悪条件を導出する際に STAMP/STPA のガイドワードを応用できる．STAMP/STPA のガイドワードは図 4-6 に示す相互作用のモデルにおいて，コントロールアクションに対して適用するガイドワードであり，境界条件の要素ごとに悪条件を導出するには抽象度が高い．そこで本研究では，図 3-7 の悪条件の概念モデルで定義した 4 つの悪条件の発生源である「ハードウェア」・「外部環境」・「オブジェクト」・「エージェントのユースケース」に対して，それぞれ具体化したガイドワードを提案する．これにより，抽象度が高い STAMP/STPA のガイドワードを使用するよりも，レビューアが BBS やユースケースから悪条件を導出しやすくなる．

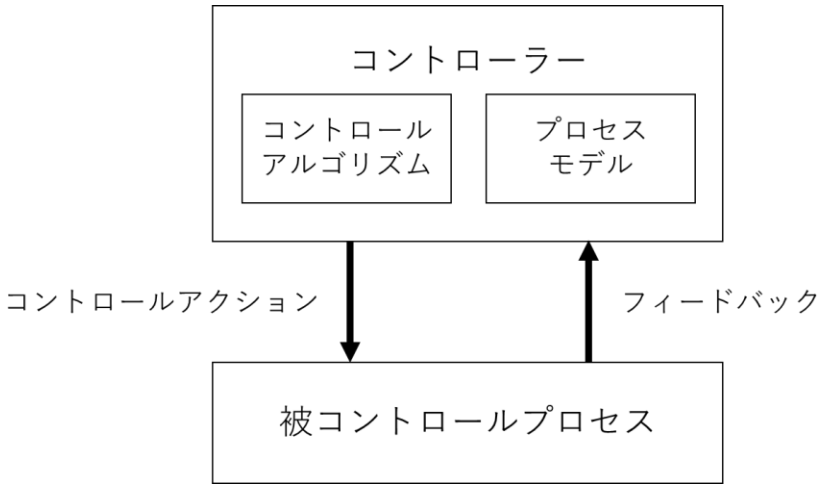


図 4-6　STAMP/STPA におけるコントロールループの一般形
※文献 [84]の図 2.6 を翻訳して引用

悪条件導出のガイドワードを構築するにあたり，まず STAMP/STPA のガイドワードを分類し直し，ガイドワードの一般形を構築した．次にガイドワードの一般形と悪条件の発生源を掛け合わせることで，ガイドワードの構造を定義した．ケーススタディで導出された具体的な悪条件をガイドワードの構造に合わせて分類・抽象化することで，ガイドワードを網羅的に定義した．

STAMP/STPA の 4 つのガイドワードを，「想定していない境界条件がある」，「想定していた境界条件の範囲を外れる」，「想定していた境界条件がない」の 3 つに分類し直し，悪条件導出のガイドワードの一般形とした．STAMP/STPA の 4 つのガイドワードと，悪条件導出のガイドワードの一般形の対応関係は表 4-5 の通りである．STAMP/STPA では相互作用のタイミングや順序に関するガイドワードと，時間の長さに関するガイドワードを分けているが，タイミング・順序や時間の長さ以外にも，境界条件の量や種類が想定範囲を超える場合など様々な悪条件が考えられることから，本研究では「想定していた境界条件の範囲を外れる」という 1 つのガイドワードにまとめた．

表 4-5　悪条件導出のガイドワードの一般形と STAMP/STPA のガイドワード

悪条件導出のガイドワードの一般形	STAMP/STPA のガイドワード
(1) 想定していない境界条件がある	与えられるとハザード
(2) 想定していた境界条件の範囲を外れる	早すぎ，遅すぎ，誤順序でハザード / 早すぎる停止，長すぎる適用でハザード
(3) 想定していた境界条件がない	与えられないとハザード

ガイドワードの一般形 3 つに対して，「ハードウェア」・「オブジェクト」・「外部環境」・「エージェント」の 4 つの悪条件の発生源を掛け合わせて表 4-6 の構造を定義した．ケーススタディで導出された具体的な悪条件をガイドワードの構造に合わせて分類・抽象化することで，表 4-6 に示すガイドワードを定義した．外部環境については，仕様で想定していない環境因子であっても，システムの周囲に常に存在はしているため，「(3) 想定していた境界条件がない」に関するガイドワードは規定しなかった．それ以外の 11 パターンの「ガイドワードの一般形」と「悪条件の発生源」の組み合わせに対して，ガイドワードを定義できた．

提案手法でレビューを実施する際，レビューアは表 4-6 のガイドワードを参考にして，BBS およびユースケースから悪条件を導出できる．例えば，前節のパン自動会計システムの事例に対してガイドワードを適用して悪条件を検討する場合，ハードウェアの「動きを邪魔される」というガイドワードからは，何らかの物体によってハードウェアの機能が阻害されると考えることで，表 4-3 の「カメラの前の障害物に撮影を邪魔される」という悪条件を導出できるほか，「操作用タッチパネルに指以外の物体が触れてタッチ操作に反応しない」という悪条件を新たに導出できる．「動きを間違える」というガイドワードからは，ハードウェアの本来あるべき動作とは異なる動作を考えることで，表 4-3 の「カメラのピントが合わず画像がぼやける」という悪条件を導出できるほか，「操作用タッチパネルのタッチした位置とは異なる位置が反応する」という悪条件を新たに導出できる．「動きが遅い」というガイドワードからは，ハードウェアの動作が本来の速度よりも遅くなるケースを想定することで，表 4-3 の「画面表示が遅い」という悪条件を導出できるほか，「カメラのピントが合うのに時間がかかる」という悪条件が新たに考えられる．「動かない」というガイドワードからは，ハードウェアが動かない状況を考えることで，表 4-3 の「カメラがフリーズする」「カメラが故障する」「本体との接続が切断される」という悪条件を導出できる

ほか,「電源の供給が断たれる」という悪条件を新たに導出できる. 本節で示した悪条件は一例であり, ガイドワードを利用すると誰もが同じ悪条件を導出できるわけではない. しかしながら, 仕様書の設計者が想定していなかった悪条件をレビューアが検討する際に, ガイドワードが思考のきっかけとなり, 多様な悪条件を想定できる.

表 4-6　悪条件導出のガイドワード

悪条件の発生源	BBS またはユースケースの要素	ガイドワードの一般形	ガイドワード
ハードウェア	入力インタフェース,名[1, 2, …, n], 出力インタフェース名[1, 2, …, n]	(1)	動きを邪魔される, 動きを間違える
		(2)	動きが遅い
		(3)	動かない
オブジェクト	オブジェクト名[1, 2, …, n], オブジェクトの属性[1, 2, …, n]	(1)	想定外のオブジェクトがある, 既定にないオブジェクトの属性の影響を受ける
		(2)	オブジェクトの属性が規定の範囲を超える
		(3)	想定したオブジェクトがない
外部環境	環境因子[1, 2, …, n]	(1)	規定にない環境条件の影響を受ける
		(2)	規定の環境条件の範囲を超える
		(3)	-
エージェント（ユースケース）	ステークホルダーの属性[1, 2, …, n], 主目的[1], 副目的[1, 2, …, n], 場所の名前[1, 2, …, n], 時間[1, 2, …, n], ステークホルダーのアクション[1, 2, …, n]	(1)	アクションを間違える, 規定にない環境条件での利用, 想定外の目的での利用, 属性に固有のステークホルダーの行動, 属性によって行動を阻害される
		(2)	アクションを繰り返す, アクションをやり直す, 目的を達成するための想定外の利用方法
		(3)	必要なアクションをしない

5. 結言

本章では提案手法のケーススタディとして, ボールペンの要求仕様書およびパン自動会計システムの要求仕様書を対象に BBS の作成, ユースケースの特定, 悪条件の導出の事例を示した. タイプの違う 2 つの製品を対象にケーススタディを行い, ハードウェア部品のみで構成される製品にも, ソフトウェアが組み込まれた製品にも提案手法を

適用可能なことを示した．STAMP/STPA のガイドワードを元に構築した 3 つの「ガイドワードの一般形」と 4 つの「悪条件の発生源」を組み合わせることで，悪条件の発生源ごとに適用可能な悪条件導出のガイドワードを構築した．ガイドワードを利用することで，ケーススタディで導出された悪条件のみならず，新たな悪条件を導出できることを示した．提案手法を利用して要求仕様書のレビューを実施する際，レビューアがガイドワードを元に悪条件を検討することで，ガイドワードなしにレビューア個人の思考だけで悪条件を検討する場合よりも広い視野で悪条件を検討できる．悪条件導出のガイドワードを含む提案手法の実際の効果については，実験による検証の結果を次章で示す．

第５章　実験による提案手法の有効性の検証

1. 序言

本章では，本研究で提案する「悪条件分析に基づく要求仕様レビュー技法」の有効性を検証するために 2 つの実験を行った結果を報告する．レビュー技法の有効性を評価するには，レビューアがレビュー技法を使い実際にレビューを実施し，その結果をもとに評価する必要がある．まず提案手法単体として，本研究で提案する境界条件と悪条件の分析のアプローチが正しく機能することを，1 つ目のレビュー実験により検証する．次に，従来のレビュー技法と比較した際の提案手法の有効性を，2 つ目のレビュー実験により検証する．

1 つ目の実験では提案手法のアプローチが正しく機能すること，すなわち提案手法のプロセスおよびフレームワークを利用して要求仕様レビューを行うことで，レビューアの考慮範囲が広がり，想定外の悪条件に起因する仕様の問題点を指摘として導出可能であることを検証する．レビューの未経験者である学生 6 名を被験者とし，アドホックレビューと提案手法を用いたレビューを実施し，その結果を比較する．提案手法はレビューの未経験者でも効果を得られ，フレームワークやガイドワードを利用することで境界条件の考慮漏れに気づきやすくなり，アドホックレビューと比較して製品の改善に繋がる有効な指摘を多く導出できることを確認する．

2 つ目の実験では提案手法を用いることで従来手法より効果の高いレビューを実施できることを検証する．製品検証の実務に携わるエンジニア 14 名を被験者とし，被験者を 2 グループに分けてクロスオーバー試験によって，従来のレビュー技法と提案手法の効果の比較を行う．従来のレビュー技法と比較して，提案手法のほうが製品の改善に繋がる重要度の高い指摘を多く導出できること，欠陥修正コストから試算したレビューの費用対効果が高いことを確認する．さらに，被験者へのアンケートおよびインタビューに基づき，境界条件と悪条件を分析するアプローチの有効性を被験者が理解し，実務でレビュー技法を活用したいと考えるか確認する．

5.2 節で 1 つ目の実験について述べ，5.3 節で 2 つ目の実験について述べる．

2. 実験 1：レビュー未経験者による提案手法の適用実験

(ア)実験の目的と概要

提案手法を要求仕様書のレビューに実際に利用することで，想定外の悪条件に起因する仕様の問題点を指摘として導出する効果が得られるかを検証する．提案手法を利用して指摘を導出するには，図 3-8 で示したとおり，①境界条件を整理する，②悪条件を検討する，③悪条件への対策を検討する，という 3 段階のプロセスを経る必要がある．①ではBBSのフレームワーク，ユースケースのフレームワークに沿ってレビューアが境界条件を整理することで，複数のアスペクトから境界条件をブレークダウンでき，レビューアが把握できる境界条件の範囲が広がる．②では悪条件導出のガイドワードを利用して悪条件を検討することで，レビューアが広い視点から悪条件を導出できる．③では悪条件への対策案のサンプルを利用して対策案を検討することで，多様な視点から対策案を検討できる．悪条件を検討して指摘を導出することは，設計者が認識していない境界条件まで考慮範囲を広げることに繋がり，製品仕様の改善に繋がる指摘が多く導出されると考えられる．

上記の提案手法の効果が機能するか確認するために，本実験では以下の 3 点を検証する．

(1) レビュー経験のない被験者でも提案手法のプロセスを経て，要求仕様書への指摘を導出できること

(2) 提案手法が持つフレームワークとガイドワードが有効に機能すること．すなわち，定型のプロセスや思考のガイドを持たないアドホックレビューでレビューした場合と比較し，提案手法を使用したほうがレビューアの考慮範囲が広がること．

(3) 提案手法を利用することで，製品の改善に繋がる重要度の高い指摘を多く導出できること

レビューの経験のない被験者にレビューという行為を体験してもらうため，最初にアドホックレビューでのレビューを実施してもらった．その後，提案手法の訓練を行い，提案手法を利用して再度レビューを実施してもらった．アドホックレビューと提案手法によるレビューで被験者の思考範囲がどの程度変化したかを確認するため，同一の被験者が連続してアドホックレビューと提案手法によるレビューを実施することとした．実験の被験者として，T 大学工学部の学部 4 年の学生 3 名，修士課程 1 年の学生 2 名，修士課程 2 年の学生 1 名の合計 6 名に協力を得た．

前述の(1)に関しては，各被験者が提案手法により指摘を導出できたことを確認するとともに，アドホックレビューと提案手法によるレビューで導出した指摘の件数を比較し，提案手法がアドホックレビューに劣らずに利用可能であることを確認する．(2)に関しては，レビュー実施時に悪条件をどの程度考慮したかについて被験者にアンケートを実施し，アドホックレビューと提案手法で考慮範囲にどの程度の差が出たかを確認する．(3)に関しては，アドホックレビューと提案手法で導出された指摘を重要度別に分類し，重要度別の指摘件数を比較することで，製品の改善に繋がる重要度の高い指摘を多く導出できたかを確認する．

(イ)実験方法

実験のタイムテーブルを表 5-1 に示す．まず被験者にレビュー対象の仕様書を渡し，特別なレビュー技法の訓練はせず，アドホックレビューによりレビューを実施してもらった．筆者らにより提案手法の訓練を実施した後，提案手法を用いて改めてレビューを実施してもらった．アドホックレビューと提案手法によるレビューともに，各被験者が個人でレビューを実施し，要求仕様書への指摘の一覧表を最終成果物として作成した．提案手法によるレビューの実施完了後，レビュー時の考慮範囲を問うアンケートに回答してもらった．

表 5-1　レビュー実験のタイムテーブル

実施内容	レビュー対象	時間
アドホックレビューによるレビュー実施	パン自動会計システムの仕様書	50 分
提案手法の訓練	ボールペンの仕様書	140 分
提案手法によるレビュー実施	パン自動会計システムの仕様書	170 分
アンケート回答	-	約 20 分

レビュー対象の仕様書はレビュー実施用と訓練用の 2 種類を用意した．いずれの仕様書も，実際の製品を参考に筆者らが実験用に作成したもので，日本語による要求仕様の記述と製品の外観や画面イメージ等の図から構成される．レビュー実施用の要求仕様書は「パン自動会計システム」の要求仕様書である．このシステムはパン屋のチェーン店における会計業務の効率化を目的とし，画像認識によりパンの種別を自動識別

して料金をレジへ自動入力する. 日本語での要求事項の記述が約 2,300 字, 画面イメージが 15 枚, 画面の一覧表および画面遷移図が各 1 個, 表紙および更新履歴が各 1 ページから構成され, 全体としては A4 サイズで 14 ページだった. 訓練用の要求仕様書は, 仮想の 4 色ボールペンの要求仕様書である. 日本語での要求事項の記述が約 1,100 文字で, A4 サイズで 1 ページに収まる分量だった. いずれの要求仕様書も意図的に欠陥を埋め込んだものではないが, レビューを行っていない状態のため, レビューにより何らかの指摘が導出されうる状態だった. 実際に実験で使用したレビュー実施用の仕様書は Appendix A に掲載した.

本実験は, 同一の被験者がアドホックレビューと提案手法によるレビューを実施することで合計 2 回のレビューを実施するため, 1 回目と 2 回目のレビューに同一の仕様書を利用すると, 2 回目のレビューでは学習効果によりレビューの結果が良くなる可能性が考えられる. しかし, レビューにおいて考慮すべき境界条件の構造を知らないレビューアは, アドホックレビューでは考慮可能な境界条件の範囲が限られており, ある程度の時間で指摘が出尽くし, アドホックレビューを2回繰り返して行っても指摘の導出件数が大きく増えることはない. 2 回目のレビューに提案手法を用いることで 1 回目のアドホックレビューよりも多くの指摘を導出できれば, 提案手法が指摘の導出件数にプラスの影響を与えたことを確認できる. また, レビューの実施経験がなくレビューの方法を知らないレビューアは, 重要度が高い指摘を狙って導出することは難しいため, 提案手法を利用することで重要度が高い指摘を多く導出できれば, 提案手法には重要度の高い指摘を効果的に導出する効果があることを確認できる.

レビュー実施の時間は, 被験者が指摘を出し尽くすまでにかかる時間を考慮して決定した. 文献[38]では, A4 用紙で 4 ページ程度の日本語による仕様書を用いたレビューの実験において, アドホックレビューで全員が十分に問題を検出できたと判断するまでレビューを続けた結果, 20 分程度でレビューを打ち切ったと報告されている. パン自動会計システムの要求仕様書は表紙・変更履歴・ページ内の空白を除くと A4 用紙で 11 ページ程度であること, 図が 7 ページ以上を占め日本語の文章を読むより時間がかからないことから, 本研究のアドホックレビューの実施時間は文献[38]の実験の 2.5 倍の 50 分とした. 提案手法によるレビューでは, BBS およびユースケースの作成・悪条件の導出の工程を経てから指摘を導出するため, アドホックレビューに比べて多くの時間が必要となることが予想された. 増えた工程にかかる時間を 120 分程度と想定し, 提案手

法によるレビューはアドホックレビューよりも 120 分長い 170 分とした.

提案手法によるレビューを実施する際は, 図 3-5 の BBS のフレームワーク, 図 3-6 のユースケースのフレームワーク, 表 4-6 の悪条件導出のガイドワード, 表 3-2 の悪条件への対策案のサンプルを各被験者に提供し, 3 章で解説したプロセスに沿ってレビューを実施してもらった. 実際に被験者に提示したフレームワーク, ガイドワード, 対策案のサンプルを Appendix B に掲載した. BBS のフレームワークとユース―ケースのフレームワークは, 編集不可のノードに色付けして, 編集可のノードとの区別を分かりやすくした. 悪条件導出のガイドワードはレビューの際に参照しやすいように表 4-6 を分割し, BBS に関連する悪条件のガイドワードと, ユースケースに関連する悪条件のガイドワードを別々の表とした.

アドホックレビューおよび提案手法によるレビューの実施が完了した後に, 被験者にアンケートに回答してもらった. アンケートの内容と形式は, レビュー時に考慮できた悪条件の範囲を問う内容とし, 5 つの選択肢から 1 つを選択する形式とした. 具体的には, 図 3-7 において悪条件の発生源として定義した「ハードウェア」,「オブジェクト」,「外部環境」,「ユースケース(エージェント)」をアンケートの対象とし, アドホックレビューおよび提案手法によるレビューを行った際のそれぞれについて, 各発生源の悪条件を考慮したかどうかを質問した. 回答選択肢は,「明確に考慮した」,「なんとなく考慮した」,「どちらとも言えない」,「あまり考慮しなかった」,「まったく考慮しなかった」の 5 つとした. 実際に実験で使用したアンケートフォームは Appendix C に掲載した.

重要度別の指摘件数, 悪条件の発生源別の指摘件数, アンケート結果を集計し, アドホックレビューと提案手法で結果に差があるか確認するために, 平均値差の検定を行った. 検定手法は並べ替え検定を用いた. 並べ替え検定はサンプルサイズが小さい場合でも正しい p 値を確認することができるという点で推奨される検定手法である[85]. 今回の実験では被験者 6 名分のデータを分析するため, サンプルサイズが小さいことから, 並べ替え検定を採用した.

(ウ)実験結果

実験結果の概要を表 5-2 に示す. 指摘件数の平均値と中央値はアドホックレビューでは 9.7 件と 10.0 件, 提案手法では 25.7 件と 25.5 件であり, 提案手法によるレビューのほうが多くの指摘を導出できた. 指摘件数の標準偏差はアドホックレビューでは 3.6,

提案手法では 12.0 であり，提案手法は被験者による指摘件数のばらつきが大きかった．1 件の指摘を導出するのにかかった時間はアドホックレビューでは 5.2 分，提案手法では 6.6 分であり，アドホックレビューのほうが若干短かった．

表 5-2　実験結果の概要

実験結果	アドホックレビュー	提案手法
指摘件数　平均値（件/人）	9.7	25.7
中央値（件）	10.0	25.5
標準偏差	3.6	12.0
レビュー実施時間（分）	50	170
指摘 1 件の導出にかかった時間（分）	5.2	6.6

アドホックレビューと提案手法の指摘件数の分布を図 5-1 に示す．アドホックレビューの指摘件数は 10 件前後に集中しているのに対し，提案手法の指摘件数は 25 件を中心に広く分布している．提案手法を利用した場合，被験者によっては大きく指摘件数が増加する効果があったと言える．

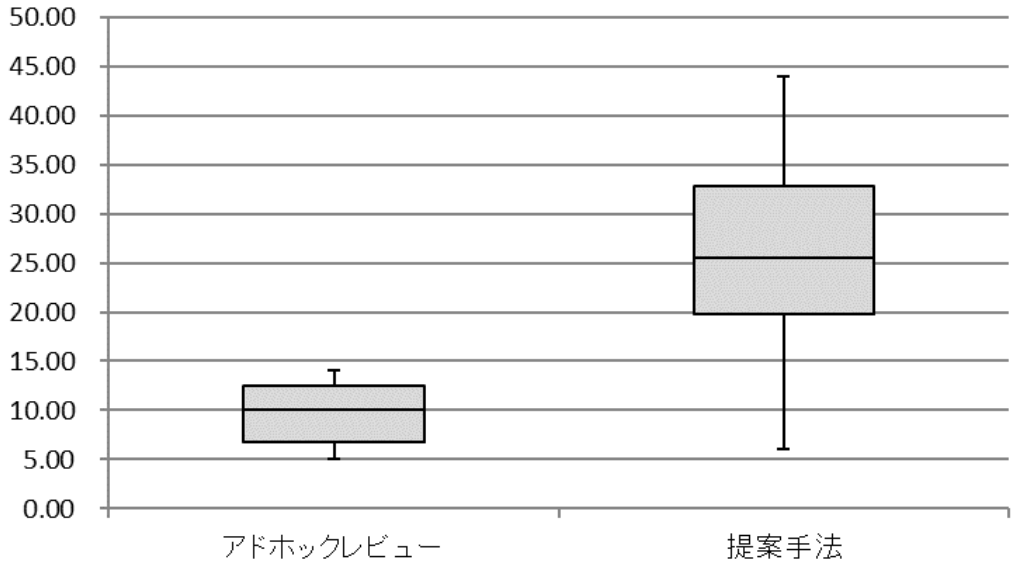


図 5-1　指摘件数の分布

導出された指摘を 3 段階の重要度で分類し，重要度ごとの平均指摘件数を集計した．重要度の定義を表 5-3 に，集計の結果を表 5-4 に示す．要求仕様書の作成時に設計者により考慮されなかった境界条件に関する指摘は，後工程で検出された場合に要求仕様自体を修正・追加する必要があり大きな手戻りに繋がることから，仕様書のレビューで検出するのが望ましい．そこで，仕様書で考慮されていなかった境界条件を考慮した改善案の重要度を「高」とした．要求仕様書で考慮されている境界条件の範囲内であっても，製品として必要な機能の定義が漏れると後工程で手戻りが発生する．ただし，詳細な機能は後工程での外部仕様書の作成時に考慮することも可能なため，製品として必要な機能の定義不足に関する指摘の重要度を「中」とした．要求仕様書で機能自体が定義されていれば詳細な動作条件や具体的な初期値や設定値等は後工程で検討可能なため，仕様書で定義されている機能の動作条件や設定値等の定義不足に関する指摘の重要度を「低」とした．

表 5-3　重要度の定義

重要度	定義
高	仕様書で考慮されていなかった境界条件を考慮した改善案
中	製品として必要な機能の定義不足に関する指摘
低	仕様書で定義されている機能の動作条件や設定値等の定義不足に関する指摘

表 5-4　重要度別の指摘件数の平均

重要度	アドホックレビュー（件/人）	提案手法（件/人）	*p* 値
高	1.5	4.3	0.078
中	3.5	11.8	0.015
低	4.7	9.5	0.216

※*p* 値は並べ替え検定により算出．

表 5-4 から，全ての重要度についてアドホックレビューに比べて提案手法の方が指摘件数の平均値は高かった．重要度が「高」の指摘はアドホックレビューの 1.5 件に対

し提案手法では 4.3 件であり約 3 倍，重要度が「中」の指摘はアドホックレビューの 3.5 件に対し提案手法では 11.8 件であり約 3 倍，重要度が「低」の指摘はアドホックレビューの 4.7 件に対し提案手法では 9.5 件であり約 2 倍となっている．重要度が「低」の指摘より，「高」「中」の指摘のほうが，提案手法を利用した際の指摘件数の増加が大きかったと言える．並べ替え検定によりアドホックレビューと提案手法の指摘件数の平均値差の検定を行った結果，重要度が「中」の指摘は $p = 0.015$ となり 5%水準で有意差があった．重要度が「高」の指摘は $p = 0.078$，「低」の指摘は $p = 0.216$ となり 5%水準で有意差はなかった．

悪条件の発生源ごとの指摘件数の平均を表 5-5 に示す．悪条件の発生源は，BBS のフレームワークで規定されている要素およびユースケースを対象とした．アドホックレビューの指摘は悪条件を明確に考慮したものではないが，指摘の内容を元にどの悪条件の発生源に該当するか筆者が判断して集計した．悪条件の発生源のうち「ソフトウェア（機能）」と「製品ライフサイクル」については，図 3-7 の悪条件の発生源には含まれていないが，BBS の「ソフトウェア（機能）」と「製品ライフサイクル」のノードに紐付けて指摘を導出した被験者が存在したため，これらに分類した．アドホックレビューにおいては，ユースケース，ハードウェア，オブジェクト，外部環境のいずれにも分類できないフィードバックを「ソフトウェア（機能）」として集計した．

表 5-5　悪条件の発生源別の指摘件数の平均

分類		アドホックレビュー（件/人）	提案手法（件/人）	*p* 値
悪条件の発生源	ユースケース	0.7	5.7	0.028
	ハードウェア	0.3	2.8	0.028
	オブジェクト	1.5	2.3	0.387
	外部環境	0.0	3.5	0.002
その他	ソフトウェア（機能）	7.2	11.2	0.493
	製品ライフサイクル	0.0	0.2	1.000

※*p* 値は並べ替え検定により算出．

ユースケースに関する指摘はアドホックレビューの 0.7 件に対して提案手法では 5.7 件，ハードウェアに関する指摘はアドホックレビューの 0.3 件に対して提案手法では 2.8 件，オブジェクトに関する指摘はアドホックレビューの 1.5 件に対して提案手法では 2.3 件，外部環境に関する指摘はアドホックレビューの 0.0 件に対して提案手法では 3.5 件，ソフトウェア（機能）に関する指摘はアドホックレビューの 7.2 件に対して提案手法では 11.2 件，製品ライフサイクルに関する指摘はアドホックレビューの 0.0 件に対して提案手法では 0.2 件となり，すべてにおいてアドホックレビューよりも提案手法のほうが指摘件数は多かった．特に外部環境の悪条件に関連する指摘はアドホックレビューでは 1 件も導出されなかったのに対し提案手法では 3.5 件導出されており，アドホックレビューで考慮できなかった境界条件に関する指摘を提案手法により考慮できたと言える．

並べ替え検定により指摘件数の平均値差の検定を行った結果，ユースケースに関する指摘は $p = 0.028$，ハードウェアに関する指摘は $p = 0.028$ であり 5%水準で有意差があった．外部環境に関する指摘は $p = 0.002$ であり，1%水準で有意差があった．オブジェクトに関する指摘は $p = 0.387$，ソフトウェア（機能）に関する指摘は $p = 0.493$，製品ライフサイクルに関する指摘は $p = 1.000$ であり，5%水準で有意差はなかった．

(エ)アンケートの集計結果

レビュー時の考慮範囲に関するアンケートの集計結果を表 5-6 に示す．アンケートの回答選択肢を「明確に考慮した＝5 点」，「なんとなく考慮した＝4 点」，「どちらとも言えない＝3 点」，「あまり考慮しなかった＝2 点」，「まったく考慮しなかった＝1 点」として被験者の回答結果を点数に置き換え，平均値を算出した．

表 5-6　レビュー時の考慮範囲に関するアンケート結果

レビューの視点	アドホックレビュー	提案手法	*p* 値
ハードウェアの悪条件	2.7	4.0	0.139
オブジェクトの悪条件	2.7	5.0	0.002
外部環境の悪条件	1.8	4.5	0.002
ユースケースの悪条件	3.7	4.5	0.219

※*p* 値は並べ替え検定により算出．

ハードウェアの悪条件はアドホックレビューの 2.7 に対し提案手法は 4.0, オブジェクトの悪条件はアドホックレビューの 2.7 に対し提案手法は 5.0, 外部環境の悪条件はアドホックレビューの 1.8 に対し提案手法は 4.5, ユースケースの悪条件はアドホックレビューの 3.7 に対し提案手法は 4.5 となり, すべてのレビューの視点についてアドホックレビューに比べ提案手法のほうが点数が高かった.

並べ替え検定によりアンケートの得点の平均値差の検定を行った結果, ハードウェアの悪条件は $p = 0.139$, オブジェクトの悪条件は $p = 0.002$, 外部環境の悪条件は $p = 0.002$, ユースケースの悪条件は $p = 0.219$ となり, オブジェクトの悪条件と外部環境の悪条件は 1%水準で有意差があった.

被験者ごとの悪条件の考慮度合いを図 5-2～図 5-5 に示す. 図の縦軸はアンケートの回答結果を表す. 表 5-6 にて有意差のなかったユースケースとハードウェアでは考慮度合いに変化のない被験者が全体の半数の 3 名ずつだったが, オブジェクトと外部環境ではすべての被験者について提案手法を利用した際のほうがアドホックレビューよりも考慮度合いが増加した.

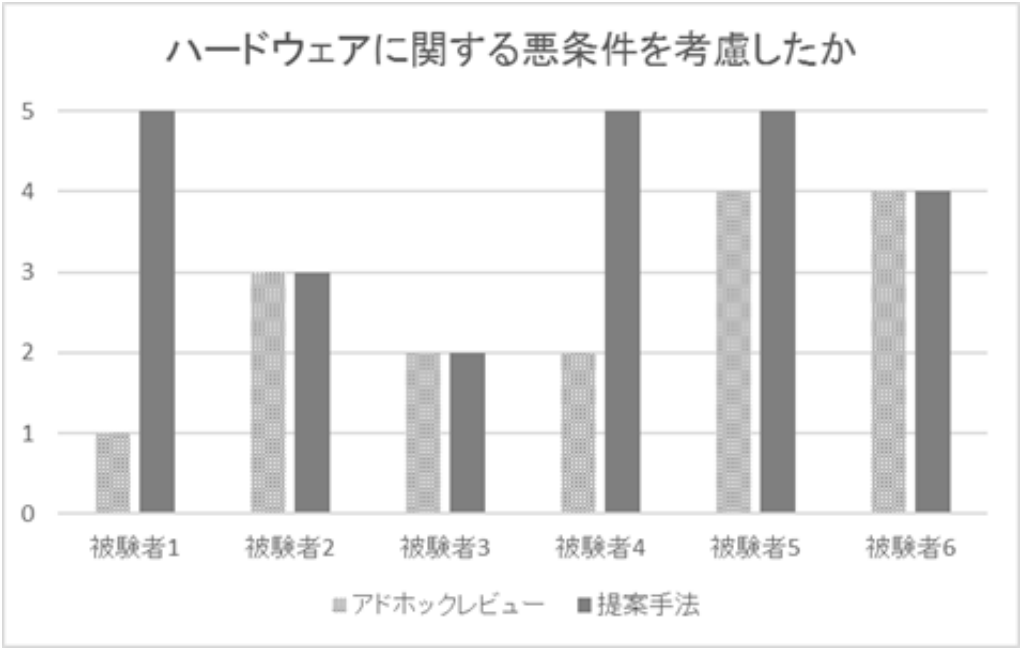


図 5-2　ハードウェアの悪条件の考慮度合い

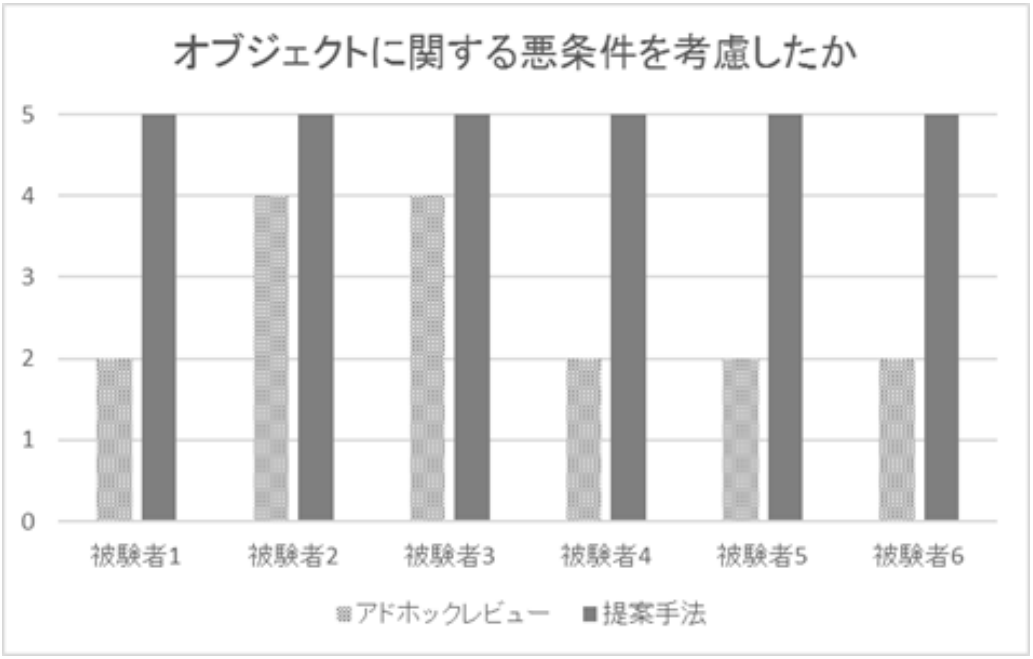


図 5-3　オブジェクトの悪条件の考慮度合い

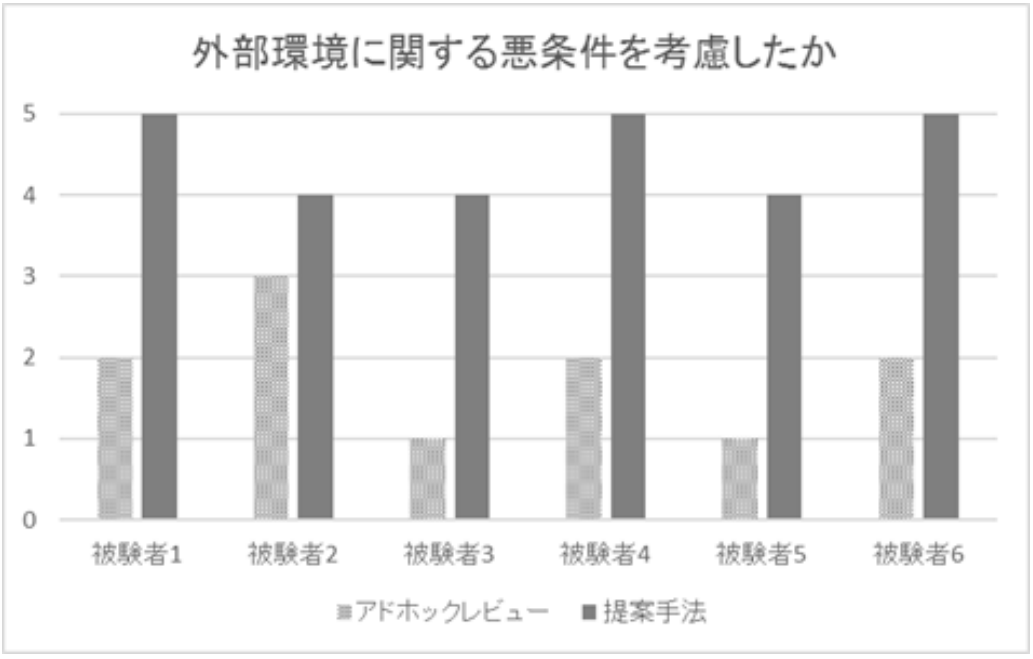


図 5-4　外部環境の悪条件の考慮度合い

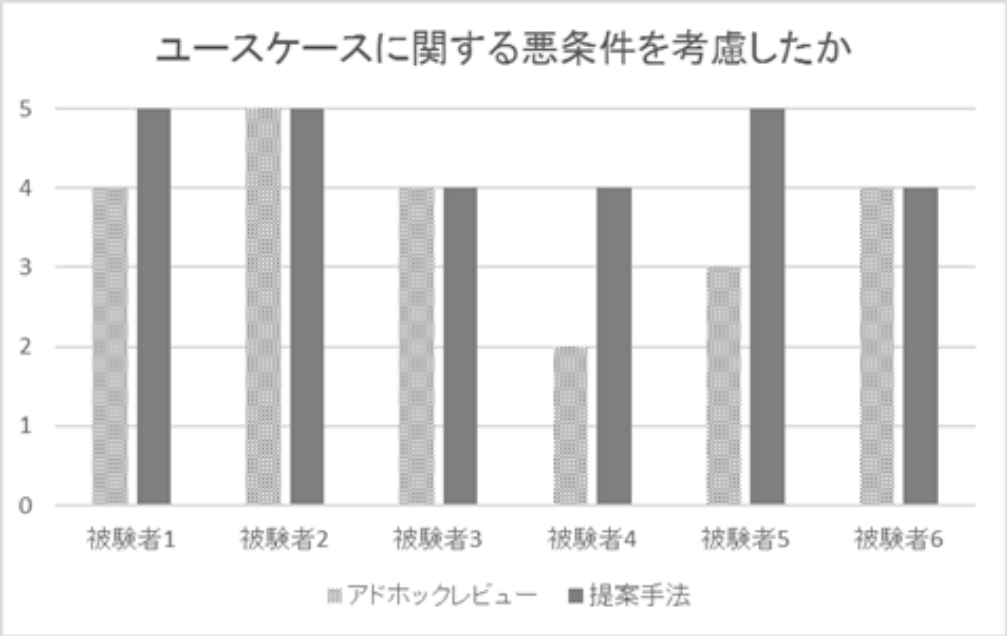


図 5-5　ユースケースの悪条件の考慮度合い

(オ)考察

実験の目的に示した「(1) レビュー経験のない被験者でも提案手法のプロセスを経て，要求仕様書への指摘を導出できること」，「(2) 提案手法が持つフレームワークとガイドワードが有効に機能すること」，「(3) 提案手法を利用することで，製品の改善に繋がる重要度の高い指摘を多く導出できること」の 3 点について，実験結果に基づき順に考察を述べる．

まず，実験の目的(1)に関して考察する．表 5-2 に示した指摘件数から，最初にアドホックレビューを実施し個人で思考できる範囲で指摘を考えつくしたにも関わらず，その後，提案手法によるレビューを実施することでより多くの指摘を導出できている．このことから，実験の目的(1)の「レビュー経験のない被験者でも提案手法のプロセスを経て，要求仕様書への指摘を導出できること」が確認できたと言える．

次に，実験の目的(2)に関して考察する．アドホックレビューと比べ提案手法によるレビューの指摘件数が増加したのは，図 5-2～図 5-5 に示したとおり，各被験者の悪条件の考慮度合いが提案手法により高まったことに起因すると考えられる．悪条件の考慮度合いが高まった要因としては，BBS およびユースケースのフレームワークを用いて境界条件を整理したこと，ガイドワードを元に悪条件を考慮したことが考えられる．

オブジェクトの悪条件と外部環境の悪条件は，表 5-6 から考慮度合いに 1%水準で有意差があった．特に外部環境については，レビュー対象の要求仕様書に明記がなくアドホックレビューで考慮できるかは被験者の思考力のみに依存していた．提案手法ではフレームワークとガイドワードを利用することで被験者が境界条件を陽に考慮できるようになり，アドホックレビューと比較して被験者全員の考慮度合いが増加したと考えられる．表 5-5 から外部環境に関する指摘はアドホックレビューでは導出されていなかったのに対し，提案手法では平均で 3.5 件導出されており，悪条件の考慮度合いの増加により指摘の導出に繋がったと考えられる．表 5-5 においてオブジェクトに関する指摘の件数に有意差がなかった理由として，要求仕様書にはパンやトレイといったオブジェクトが記載されていたため，明確に境界条件を意識せずともアドホックレビューでもオブジェクトに関する指摘を導出できていたことが挙げられる．アドホックレビューでは仕様書の記載の有無に影響を受け，仕様書に境界条件の記載がある場合は，陽に悪条件を考慮しなくても指摘を導出できる場合があると言える．仕様書の記載の有無にレビュー結果が影響を受けると，仕様書に記載されていない想定外の外的条件の考慮が漏れるため，フレームワークやガイドワードを元にして境界条件を整理するという提案手法のアプローチは，仕様書の記載があいまいな場合に特に有効であると考えられる．

ハードウェアの悪条件とユースケースの悪条件は，表 5-6 から考慮度合いに有意差はなかった．ハードウェアに関しては，ディスプレイやキーボードといったハードウェアが要求仕様書に記載されていたため，アドホックレビューにおいても被験者が考慮しやすかったと考えられる．ユースケースに関しては，レビューを行う際は通常，ユーザがどのように製品を利用するかを想定して指摘を導出するため，アドホックレビューにおいても被験者が自然と考慮していたと考えられる．表 5-5 からハードウェアに関する指摘，ユースケースに関する指摘ともに，アドホックレビューに比べて提案手法のほうが指摘件数が増加しており，悪条件の考慮度合いが同程度であっても，悪条件の導出にガイドワードを利用する提案手法のほうが，より多くの指摘を導出できている．ガイドワードを利用して悪条件を考慮することが，より多くの指摘の導出に繋がっていると言える．アドホックレビューでユースケースに関する指摘が少なかった理由として，アドホックレビューではどのようなユースケースを考慮してレビューしたかが記録として残らないため，被験者がユースケースを考慮して指摘していたとしてもレビュー結果を分類した際にユースケースに関する指摘として分類されなかった可能性が挙げられる．アドホックレビュー

の指摘は，レビュー実施後に第三者が内容を確認しても，ユースケースを考慮した指摘かどうかが明確には分からない．このような状態では必要十分なユースケースが考慮されたか判断するすべがない．レビューアが考慮した内容を明確に表現し，ステークホルダー間での共有や議論が可能になるという点でも，提案手法におけるユースケースのフレームワークを利用する価値があると考える．

以上の結果から，レビュー対象の仕様書における境界条件の記載の有無に関わらず，提案手法のフレームワークを利用することでレビュー時に考慮すべき境界条件をレビューアが明示的に考慮でき，ガイドワードを利用することで悪条件および指摘を多く導出できることが確認できた．実験の目的(2)の「提案手法が持つフレームワークとガイドワードが有効に機能すること」が確認できたと言える．

最後に，実験の目的(3)に関して考察する．表 5-4 から，アドホックレビューに比べて提案手法では，すべての重要度で指摘件数が増えていることに加え，重要度が「低」の指摘よりも「高」「中」の指摘の割合が増加している．提案手法のフレームワークを利用することでアドホックレビューでは考慮できなかった境界条件を考慮できるようになる点と，ガイドワードを利用することで多くの悪条件を導出できるようになる点から，要求仕様書で定義が漏れていた境界条件や製品として必要な機能に気付くことが可能となり，重要度が「中」以上の指摘の割合が増加したと考えられる．本実験の目的(3)の「提案手法を利用することで，製品の改善に繋がる重要度の高い指摘を多く導出できること」が確認できたと言える．

本節の実験により，提案手法はレビューの未経験者でも効果を得られ，フレームワークやガイドワードを利用することで境界条件そのものの考慮漏れに気づきやすくなり，アドホックレビューと比較して製品の改善に繋がる有効な指摘を多く導出できることが確認された．提案手法は要求仕様書のレビュー技法として一定の効果があると考えられるが，本節の実験は(1)比較対象が特別な方法を利用しないアドホックレビューである，(2)レビューに投入する時間が実験ごとに異なり，単純に指摘事項の件数を比較できない，(3)同一の被験者がレビューを繰り返し実施しており学習効果の影響を排除できないという課題がある．提案手法の効果が従来のレビュー技法と比較した場合にどれほど優位なのかは検証できていない．

提案手法の課題として，被験者による指摘件数のばらつきが大きく，レビューの結果がレビューアの能力に依存しやすいことが挙げられる．提案手法によるレビューは，構

造化された BBS・ユースケースのフレームワークや悪条件の検討のためのガイドワードを用いることから，それらを用いない場合に比べて指摘件数が増加するのは一見すると当然のように見える．しかしながら，図 5-1 の結果から，提案手法を用いた場合の指摘件数がアドホックレビューでの平均指摘件数よりも少ない被験者がいることが分かった．提案手法の効果を得るには，境界条件を階層構造でモデリングする能力，BBS のフレームワークや悪条件導出のガイドワードなど，抽象的な思考のガイドを元にして境界条件や悪条件を想定する能力がレビューアに必要となる．提案手法の訓練を行ったからといって，すぐにレビューアのこれらの能力が向上するわけではない．結果として，表 5-2 および図 5-1 に示したように被験者による指摘件数のばらつきが大きかったと考えられる．レビューアの能力にレビューの結果が依存するという点は，従来手法においても同様であり，文献[86]ではレビューの指摘件数は開発経験年数などの指標との相関が低く，サンプルのレビュー対象に対して実施したレビューの指摘件数との相関が高いとの報告があり，指摘を多く出せる被験者はどんな方法を使っても指摘を多く出せる傾向があると予想される．文献[43]の実験ではアドホックレビューより PBR のほうが欠陥検出率のばらつきが大きいという結果が出ており，特定のレビュー技法を利用すると，多くの指摘を検出できるレビューアはより多くの指摘を検出でき，結果として，それほど多くの指摘を検出できないレビューアとの指摘件数のばらつきが大きくなると考えられる．提案手法は PBR よりもさらにばらつきが大きくなったことから，従来手法よりも指摘の導出効果が高いかは自明でない．

提案手法のもう一つの課題として，BBS の作成・ユースケースの作成・悪条件の導出といった多くの中間成果物を作成する必要があり，レビューの実施に時間を多く要する点が挙げられる．表 5-2 に示したとおり，提案手法によるレビュー実施時間はアドホックレビューと比較して少なくとも 3 倍以上が必要だった．時間がかかったとしても多くの指摘を導出できれば費用対効果は高くなるが，従来手法よりも費用対効果が高いかは自明でない．

これらの課題を踏まえ，次節では，クロスオーバー試験に基づく実験計画を立案し，従来のレビュー技法で最も効果的とされる PBR と提案手法の比較実験を行う．従来手法と比べ，提案手法のほうが指摘導出の効果および費用対効果が高いかを検証する．

3. 実験 2：クロスオーバー試験に基づく従来手法との比較実験

(ア)実験の目的と概要

本節では提案手法の有効性を検証するため，従来のレビュー技法との比較実験を行う．前節の実験により，提案手法のフレームワークとガイドワードを用いることでレビューアの考慮範囲が広がり，製品の改善に繋がる重要度の高い指摘を多く導出できることが確認された．しかし，前節の実験では比較対象がアドホックレビューであること，レビュー実施の時間がレビュー技法ごとに異なり指摘件数を単純に比較できないこと，同一の被験者がレビューを繰り返し実施しており学習効果の影響を排除できないという課題があり，提案手法の有効性を十分に評価できたとは言えない．本節では，これらの課題を解消し従来のレビュー技法と比較した場合の提案手法の有効性を明らかにするために，クロスオーバー試験という試験法を導入し比較実験を行う．

クロスオーバー試験は少数の被験者により 2 つの手法の効果を検証する試験方法として標準的に確立された試験方法である [87]．クロスオーバー試験は新薬の効果の比較などエビデンス重視の臨床研究で古くから応用されており，文献[88]でさまざまなクロスオーバー試験が紹介された．文献[89]では牧畜業の分野での適用例の報告がある．クロスオーバー試験は，確率を用いて実験対象の群を分ける「ランダム化試験」と呼ばれる試験法の一種である．ランダム化試験は臨床研究のさまざまなガイドラインにおいてエビデンスレベルが高い試験法とされる[90][91]．文献[92]はエビデンスレベルを階層構造で整理しており，クロスオーバー試験は研究レビューに次いでエビデンスレベルが高いとしている．研究レビューは文献[93]で言及され，ランダム化試験による複数の研究の結果を統合して最終的な結果を論じる研究手法である．本研究では単一の比較実験により提案手法の有効性を検証するために，ランダム化試験の一種であるクロスオーバー試験を採用した．クロスオーバー試験により手法の効果検証を行うには，被験者を 2 グループに分け，各グループで手法を使用する順番を入れ替えて 2 つの手法を順に使用する．同一の被験者の各個人内で結果を比較でき，単純な 2 グループの比較試験よりも結果のバラツキが小さくなり，少数の被験者で実験可能となる[94]．どのレビュー技法を使ったとしても人間の思考によりレビューが行われることは共通しており，レビューの結果が被験者の能力に依存することを避けるのは難しい．クロスオーバー試験を用いず単純な 2 グループの比較試験を行った場合，結果に差が生まれたとしても比較すべきレビュー技法の効果によって差が生まれたのか，各グループの被

験者の能力によって差が生まれたのか結論付けることはできない. それに対し, クロスオーバー試験では順序を入れ替えて各グループともに 2 つのレビュー技法を順に用いることで, 被験者の能力差による要因を排除できる. 本研究ではレビュー技法の効果検証にクロスオーバー試験を用いることで, 被験者の能力差に依らない形で, 2 つの手法の効果を適切に比較可能とした.

従来のレビュー技法は欠陥検出のシナリオに基づいてレビューアが仕様書を読むことで網羅的に欠陥を検出する効果がある[95]. 既定のシナリオに沿って仕様書を読むという従来のレビュー技法のアプローチと比べて, 境界条件の整理と悪条件の検討を行うという提案手法のアプローチは, レビューアの考慮範囲を広げ想定外の外的条件に起因する問題を検出しやすくなることが期待される. しかし, 前節の実験ではアドホックレビューを比較対象としているため, 従来のレビュー技法と比較して提案手法のほうが, 製品の改善に繋がる重要度の高い指摘を多く導出できるかは明らかでなく, 検証が必要である.

提案手法は BBS や悪条件一覧等の中間成果物を作成する必要があることから, 従来のレビュー技法よりもレビューの実施に多くの時間がかかる. 前節の実験では, 提案手法はアドホックレビューの 3 倍強の時間を要した. 重要度の高い指摘を多く導出できれば, 同じ欠陥を製品開発の後工程で検出した場合に比べて欠陥の修正コストが低くなるため, レビューに時間をかける価値がある. 欠陥修正コストの低減の効果はレビューの有効性を測る重要な指標であり, 提案手法の有効性を検証するためには, 従来のレビュー技法と提案手法で欠陥修正コストの低減の効果を比較する必要がある.

前節の実験により, レビューの未経験者であっても 2.5 時間程度の訓練で提案手法を利用可能になることが確認できた. しかし, 従来のレビュー技法と比べると提案手法は手順や中間成果物が多く, 実施の難易度が高いことや実施にかかる時間が長いことが実用上の課題になりうる. 従来のレビュー技法と提案手法の両者を利用した実務者に, 各レビュー技法を実務で活用可能かを評価してもらい, 提案手法の実用可能性が従来のレビュー技法と比較して問題ないことを確認する必要がある.

以上から, 本節の実験では以下の 3 点を確認する.

(1) 要求仕様書の記述の詳細さに関わらず, 提案手法によりレビューアの考慮範囲が広くなり, 製品の改善に繋がる重要度の高い指摘を従来のレビュー技法より多く導出できるか

(2) 費用対効果が従来のレビュー技法と比較して高いか

(3) 製品検証の実務に携わるエンジニアが提案手法の効果を認め実務で活用したいと思うか

仕様記述の詳細度が異なる 2 種類の仕様書を用いて従来のレビュー技法と提案手法でレビューを行い，導出された指摘の件数とアンケートおよびインタビューの回答結果に基づき上記の 3 点を確認する．(1)については，レビュー時に悪条件をどの程度考慮したかを被験者へのアンケートで確認するとともに，被験者が導出した指摘を重要度別に集計し，従来のレビュー技法と提案手法の結果を比較する．一人の被験者が類似の指摘を多数導出した場合，単純に指摘件数を比較してもレビュー技法の効果を適切に評価できない可能性があるため，指摘を類似の内容ごとに分類し，分類の件数も比較する．指摘の分類のうち，各レビュー技法で導出できたものの割合を集計し，提案手法のほうが多くの指摘を網羅できているか確認する．(2)については，文献[96]にてレビューの効果を金額に変換する方法が提案されており，この方法を用いて，レビュー実施にかけたコストに対する費用対効果を算出する．(3)については，製品検証の実務に携わるエンジニアを本実験の被験者とし，提案手法と従来のレビュー技法を利用して要求仕様書のレビューを実施した後，アンケートおよびインタビューにより提案手法を実務で活用したいと思ったか，提案手法の効果をどのように感じたかを確認する．

クロスオーバー試験の形式に基づく実験全体の流れを図 5-6 に示す．被験者を 2 つのグループに分け，1 回目と 2 回目で異なるレビュー技法を使用してレビューを実施してもらい，1 回目・2 回目それぞれのレビュー結果を比較する．

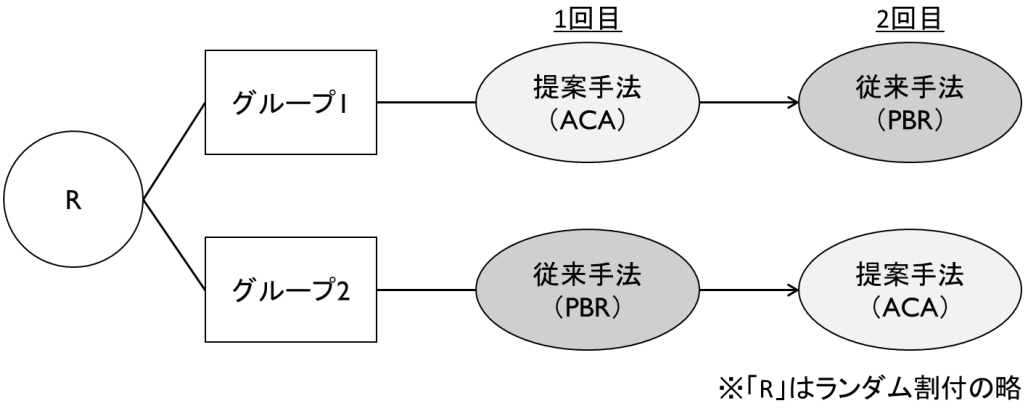


図 5-6　実験全体の流れ

提案手法と比較を行う従来のレビュー技法は Perspective-based reading（PBR）とした. ISO/IEC20246:2017 [22]では PBR が「最も効果的かつ効率的なレビュー技法」とされることから, レビュー技法の効果を比較する対象として適切だと考えた. PBR の実施方法には境界条件や悪条件の分析のプロセスは明示されていないが, PBR は欠陥を網羅的に検出する手法であり, 想定外の悪条件に起因する仕様の欠陥も検出の対象になる. そのため, PBR によるレビューの結果は提案手法によるレビューの結果と比較することが可能である.

実験の被験者として V 社のエンジニア 14 名に協力を得た. V 社はシステム・製品検証の専門会社であり, その実務ではシステムや製品の検証の過程で仕様の不備の指摘を行っており, 実験の被験者として適切と言える. 過去に行われた PBR と他の技法との比較実験では, 時間およびコストの制約を考慮すると被験者は 12～18 名という人数が妥当であると報告されている[42]. 14 名という本実験の被験者の人数は過去の研究と同等であり, 2 つのレビュー技法の比較実験を行うにあたって問題ない人数であると言える.

なお文献[42]における PBR の効果検証の実験はクロスオーバー試験の形式ではなかった. 本論文の実験方法と文献[42]の実験方法を表 5-7 に整理した. 文献[42]の実験方法では, 2 グループともレビュー技法の使用順序は同じとなっている. 1 回目のレビューでは比較対象のレビュー技法を用い, その後の 2 回目のレビューでは PBR を使用してレビューを行っている. 表 5-7 に矢印で示したとおり, この実験方法で実験を行った場合はグループ間で交差する形でレビュー結果を比較する必要がある. この実験方法では 2 つの技法を知った状態でレビューを行ったグループと, 1 つの技法しか知らない状態でレビューを行ったグループの間で結果を比較する形になる. 1 回目・2 回目の各レビューでその時使用するレビュー技法の知識以外に, もう一方のレビュー技法に関する知識をレビューアが持っているか否かという差が生まれ, この差がレビュー結果に予期しない影響を与える可能性がある. これに対して本研究は, 従来のレビュー技法の研究では実施されていなかったクロスオーバー試験を, レビュー技法の効果の比較のために導入した. 表 5-7 に矢印で示したとおり, クロスオーバー試験ではグループ 1 とグループ 2 の間で, 1 回目のレビューと 2 回目のレビューのそれぞれで結果を比較する形となる. この形ではグループ 1・グループ 2 ともに, 1 回目のレビューではその時に使用したレビュー技法のみ知っておりもう一方のレビュー技法を知らない状態, 2 回目

のレビューではレビュー技法を 2 つとも知った状態となる. 1 回目のレビュー・2 回目のレビューともに, レビューアのレビュー技法に関する知識の差がなく, 各試行で実際に使用したレビュー技法のみが異なる状態でのレビュー結果を比較できる. クロスオーバー試験を用いることで, レビューアの知識の差が結果に予期しない影響を与えることを防ぐ形で評価が可能となる. レビュー技法の効果を比較するにはクロスオーバー試験の形式が適切だと言える. 方法に関する研究の正しさを主張することは一般的に難しいが, この分野で従来は実施されていなかったクロスオーバー試験を導入することで, 本研究ではレビュー技法の効果を適切に検証可能とした.

表 5-7　本研究と文献[42]の実験の比較

レビュー	本研究			文献[42]		
回数	グループ 1		グループ 2	グループ 1		グループ 2
1 回目	提案手法	↔	従来手法	Usual technique		Usual technique
	仕様書 1		仕様書 1	仕様書 A	✕	仕様書 B
2 回目	従来手法	↔	提案手法	PBR		PBR
	仕様書 2		仕様書 2	仕様書 B		仕様書 A

14 名の被験者を 7 名ずつの 2 グループに分けて実験を行った. 各被験者は個人で仕様書のレビューを行い, 仕様書への指摘事項を導出した. システム・製品検証に関する被験者の業務経験年数は 1 年～23 年だった. 業務経験年数の平均値, 中央値, 標準偏差をグループごとに算出した結果を表 5-8 に示す. 標準偏差はグループ 1 のほうが大きいが, 平均値・中央値はグループ 1 とグループ 2 ともに 8 年前後で同程度であり, 業務経験という点ではグループ間に大きな差はなかったと言える.

表 5-8　システム・製品検証業務の経験年数

検証業務経験年数	グループ 1	グループ 2
平均値(年)	8.1	7.7
中央値(年)	8.0	9.0
標準偏差	7.0	3.4

(イ)実験方法

実験のタイムテーブルを表 5-9 に示す. まず被験者に対して筆者らがレビュー技法の訓練を 120 分間実施した. 訓練の内容としては, 筆者らがレビュー技法の使い方を説明したうえで, 訓練用に用意した仕様書をレビュー対象とし被験者にレビューを実施してもらった. 訓練が終わった後, 実験用の仕様書を被験者に渡し, 被験者自身が指摘を出し尽くしたと思うまで個人レビューを実施してもらい, 指摘の一覧表を最終成果物として提出してもらった. レビュー開始前の説明の際に, レビューの目的を次のように口頭および紙面で被験者に伝えた.「今回のレビューの目的は, 要求仕様書がこれから製品開発を進めるにあたって妥当なものになっているかを確認することである. 要求仕様書が妥当であるとは, この要求仕様書に基づいて製品を開発するとエンドユーザにとって有用な製品が出来上がるような要求仕様書になっている, ということを表す」. このような説明を行い, 要求仕様書の内容が妥当ではなく内容の修正や再検討が必要だとレビューアが判断した部分に対して, 考えつく限りすべての指摘を導出してもらった. 1 回目のレビュー実施と 2 回目のレビュー実施ともに, レビューの実施完了後に, 悪条件の考慮度合いおよびレビュー技法の実務での活用可能性に関するアンケートに回答してもらった. 実験完了後に被験者へインタビューを行い, 提案手法について意見を得た.

表 5-9　レビュー実験のタイムテーブル

回数	実施内容	レビュー対象	時間
1 回目	レビュー技法の訓練	電気ポットの仕様書	120 分
	レビュー実施	パン自動会計システムの仕様書	レビュー完了まで
	アンケート回答	—	約 10 分
2 回目	レビュー技法の訓練	自動販売機の仕様書	120 分
	レビュー実施	腕時計型通信端末の仕様書	レビュー完了まで
	アンケート回答	—	約 10 分

レビュー対象の要求仕様書は, 同一の仕様書を 2 回使用すると学習効果による実験結果への影響が懸念されるため, 1 回目と 2 回目のレビューで異なる仕様書を用いる必要がある. さらに本実験では, レビュー技法によるレビューアの考慮範囲の差を確認

するため，1回目のレビューと2回目のレビューで特徴の異なる仕様書を用意した．1回目のレビューでは，前節の実験でも使用したパン自動会計システムの要求仕様書をレビュー対象とした．2 回目のレビューでは腕時計型通信端末の要求仕様書をレビュー対象とした．どちらの仕様書も同様に，要求仕様の日本語による記述と，画面イメージや製品の外観等の図を含んだ構成となっている．実務での仕様書のレビューに近い状況を再現するために，仕様書への意図的な欠陥の埋め込みはしなかった．腕時計型通信端末の要求仕様書は，V 社における製品検証の研修用に作成された仮想の要求仕様書である．この製品はスマートフォンと接続して音声通話やビデオ通話を行うことができ，BtoC にて販売される製品という想定である．日本語での要求事項の記述が約2000 字，外観図が 4 個，表紙および目次と更新履歴が各 1 ページであり，全体としてはA4 サイズで 7 ページだった．訓練用の仕様書は学習効果による実験結果への影響を防ぐために，実験用の仕様書とは別の仕様書を用意した．訓練用の仕様書としては，インターネット上で公開されている電気ポットの要求仕様書[97]と自動販売機の機能仕様書[98]を利用した．

パン自動会計システムの要求仕様書と腕時計型通信端末の要求仕様書の特徴の比較を表 5-10 に示す．各仕様書は，想定される製品の提供形態や利用者，要求仕様の記述の詳細度が異なった．パン自動会計システムはパン屋のチェーン店の屋内で利用されることを想定したシステムであり，想定される利用シーンが限定的なことに加え，要求仕様は詳細に定義されており情報量が多かった．腕時計型通信端末は不特定多数のユーザによる利用が想定される製品であり，利用シーンが多岐にわたることに加え要求仕様は抽象的な記述が多く，考慮すべき境界条件があいまいだった．

表 5-10　レビュー対象の仕様書の特徴の比較

特徴	パン自動会計システム	腕時計型通信端末
言語	自然言語（日本語）	自然言語（日本語）
分量	14 ページ，約 2300 字	7 ページ，約 2000 字
製品の提供形態／利用者	BtoB／パン屋の店員・顧客が利用	BtoC／不特定多数が利用
要求仕様の記述の詳細度	詳細な記述が多い	概要的な記述が多い

訓練およびレビュー実施の際，レビュー技法で利用するレビュー観点・フレームワーク・ガイドワードを被験者に対して提示した．本実験で使用した PBR のレビュー観点を表 5-11 に示す．表 5-11 は文献[43]にて「利用者／顧客」，「設計者／開発者」，「テスト担当者」の 3 つの視点で整理されたレビュー観点を元に，筆者が一部修正した．岡本らの実験では，要求仕様書の上位文書としてニーズ記述書が存在し，ニーズ記述書と要求仕様書を比較することで欠陥を検出するレビュー観点が存在したが，実際の開発現場ではニーズ記述書が存在するとは限らないため本実験ではニーズ記述書は用意せず，該当するレビュー観点は除外した．PBR は担当者ごとに異なる視点を割り当ててレビューを実施することで指摘の重複を防止するメリットがあるが，本実験では個人レビューにおいてレビューア一人ひとりが指摘した件数を提案手法と比較するため，各被験者に全ての視点のレビュー観点を使用してレビューを実施してもらった．

提案手法によるレビュー実施の際は，前節の実験と同様に Appendix B に掲載した BBS のフレームワーク，ユースケースのフレームワーク，悪条件導出のガイドワード，悪条件への対策案のサンプルを各被験者に提供し，3 章で解説したプロセスに沿ってレビューを実施してもらった．

表 5-11　PBR のレビュー観点

視点	レビュー観点
利用者／顧客	1.要求仕様書に記述されたすべての要求は，あなたがソフトウェア製品に期待する内容か？ 2.システムを利用するすべての利用者は識別されているか？ 3.すべての状況における入力に対して，ソフトウェアの振る舞いが定義されているか？ 4.未決事項の要求が含まれていないか？ 5.問題領域に固有な用語は明確に定義されているか？ 6.ソフトウェア設計や実装に関する専門用語が使われているために，あなたが理解しにくい箇所はないか？
設計者／開発者	1.設計に必要な情報（機能，性能，設計上の制約）はすべて定義されているか？ 2.すべての外部インタフェースは明確に定義されているか？ 3.すべての要求は一意にしか解釈できないように記述されているか？ 4.冗長な要求記述が含まれていないか？ 5.要求仕様書の内部で，他の記述と互いに矛盾しているために，設計すべき内容が理解できない部分があるか？ 6.要求が詳細に記述されすぎているために，あなたが理解できていない部分があるか？ 7.ソフトウェアの内部設計に関する記述が書かれてしまっていないか？ 8.そのシステムに関する知識またはシステム導入の目的や概要に書かれた内容から考えて，要求仕様は意味をなしているか？
テスト担当者	1.要求される正確性および精度が定義されているか？ 2.すべての機能が要求を満たしていることを確認する方法について，あなたは明確に理解できているか？ 3.すべての機能が正しい値を出力していることを確認する方法について，あなたは明確に理解できているか？

※文献[43]の付録 F を元に筆者が一部修正

レビュー実施後のアンケートは，レビュー時の悪条件の考慮度合いとレビュー技法の業務での活用可能性に関して，5 段階の選択肢から一つを選択する形式とした．レビュー時の悪条件の考慮度合いに関する質問は，PBR のレビューの視点である「利用者/顧客」・「設計者/開発者」・「テスト担当者」と，提案手法のフレームワークで規定された悪条件の発生源である「ハードウェア」・「オブジェクト」・「外部環境」・「ユースケース」を対象とし，「利用者/顧客の視点でレビューを行いましたか？」・「設計者/開発者の視点でレビューを行いましたか？」・「テスト担当者の視点でレビューを行いましたか？」・「ハードウェアに関する悪条件を考慮してレビューを行いましたか？」・「オブジェクトに関する悪条件を考慮してレビューを行いましたか？」・「外部環境に関する悪条件を考慮してレビューを行いましたか？」・「ユースケースに関する悪条件を考慮してレビューを行いましたか？」という質問項目を用意した．回答選択肢は「明確に考慮した」・「なんとなく考慮した」・「どちらとも言えない」・「あまり考慮しなかった」・「全く考慮しなかった」の 5 つとした．業務での利用可能性に関する質問では，「本日使用したレビュー技法を業務で活用したいと思いますか？」という質問項目を用意し，「非常にそう思う」・「そう思う」・「どちらとも言えない」・「思わない」・「まったく思わない」を回答選択肢とした．「非常にそう思う」・「そう思う」と回答した被験者には回答の理由として「重要な指摘を出せそうだから」・「レビューの時間を短縮できそうだから」・「普段思いつかない指摘を思いつくことができそうだから」・「指摘の漏れを減らせそうだから」・「指摘の途中経過を含めて顧客に説明しやすそうだから」・「その他（自由記述）」から該当するものを複数選択してもらい，「どちらとも言えない」・「思わない」・「まったく思わない」と回答した被験者には回答の理由として「難易度が高いから」・「実施するのに時間がかかるから」・「効果があるか不明だから」・「技法がなくても同様の指摘を出せそうだから」・「自分の担当する製品にこの技法を適用するイメージが沸かないから」・「技法を理解できていないから」・「その他（自由記述）」から該当するものを複数選択してもらった．実際に実験で使用したアンケートフォームは Appendix D に掲載した．

(ウ)実験結果

提案手法によるレビューの成果物の事例として，ある被験者が作成した BBS・ユースケース・悪条件・仕様への指摘の一部を図 5-7・図 5-8・表 5-12 に示す．図 5-7 に示したとおり，この被験者は，エージェントとしてはパン屋の店員である「店長」・「店員」やパン

を購入する客である「高齢者」・「子供」など，オブジェクトとしてはパンを乗せるための「トレイ」・システムを電源に接続するための「コンセント」・店員が装着する「手袋」など，外部環境としてはカメラによる撮影に影響を与える「明るさ」などをBBSのノードとして導出した．悪条件は図5-7に斜体で記載されており，「高齢者」というエージェントによる悪条件として「視力が悪く，文字が見えない」，「子供」というエージェントによる悪条件として「背が小さく，ディスプレイが見えない」，「トレイ」というオブジェクトによる悪条件として「白色以外のトレイを使った」，「コンセント」というオブジェクトによる悪条件として「コンセントの長さがギリギリで，抜けてしまう」，「手袋」というオブジェクトによる悪条件として「手袋をつけてタッチ操作する」，「明るさ」という外部環境による悪条件として「室内環境によって画面が見づらい」などを被験者が導出した．

BBSに記述されたエージェントのうち，「店員」のユースケースが図5-8である．30代の女性の店員を想定し，「パンを売る」というユースケースを被験者が作成した．パン自動会計システムの使い方として，客がレジに到着してから会計を済ませて店員がパンを客に渡すまでの一連の流れが挙げられており，「お客さんがレジに来たら，待ち受け画面をやめる」・「パンを読みとらせる」といった使い方が記載された．その他に，パンを割引価格で販売することを想定し，「パンを割引する」という使い方が記載された．悪条件は図5-8に斜体で記載されており，「お客さんがレジに来たら，待ち受け画面をやめる」という使い方に対する悪条件として「待ち受け画面を閉じる方法がわからない」，「パンを割引する」という使い方に対する悪条件として「割り引いて計算できない」などを被験者が導出した．

これらの悪条件への対策案を検討して被験者が作成した指摘一覧が表5-12である．No.1・2・6の悪条件に対しては，表示フォントの変更・ディスプレイの角度の変更・画面の輝度の調整といった，ユーザや利用環境を考慮した仕様の改善案が指摘として導出された．No.3・5の悪条件に対しては，使用可能なトレイの条件・ユーザが手袋をしている場合のタッチ操作の可否といった，仕様で考慮が不足していた利用時の制約が指摘として導出された．No.4の悪条件に対しては，急に電源が落ちた場合という異常系に関する対策案が指摘として導出された．No.7の悪条件に対しては，仕様で明記されていなかった画面遷移手順に関する改善案が指摘として導出された．No.8の悪条件に対しては，割引計算の機能を追加するというユースケースに基づく機能の追加案が指摘として導出された．被験者が様々な視点で指摘を導出できたことが分かる．

図 5-7　ある被験者が作成した BBS と悪条件の一部

図 5-8　ある被験者が作成したユースケースと悪条件の一部

表 5-12　ある被験者が作成した指摘一覧の一部

No.	悪条件	悪条件への対策のタイプ/ポイント	指摘/悪条件への対策案
1	視力が悪く，文字が見えない	悪条件の低減/システム	表示フォントを変更できるようにする
2	背が小さく，ディスプレイが見えない	悪条件の低減/システム	ディスプレイの角度を変更できるようにする
3	白色以外のトレイを使った（※注）	悪条件の回避/ユーザ	認識可能なトレイの色，サイズを取扱説明書等でユーザに通知する
4	コンセントの長さがギリギリで，抜けてしまう	悪条件の回避/システム	急に電源が落ちてもシステムに影響が無いようにする
5	手袋をつけてタッチ操作する	悪条件の低減/ユーザ	手袋をしていると画面のタッチ操作が行えないのであれば，その旨を取扱説明書等で，ユーザに通知する
6	室内環境によって画面が見づらい	悪条件の低減/システム	画面の輝度を調整できるようにする
7	待ち受け画面を閉じる方法がわからない	悪条件の回避/システム	待ち受け画面表示中に，何らかのキーを押下したら，準備画面が閉じるようにする
8	割り引いて計算できない	悪条件の回避/システム	計算時に「xx%OFF」といった割引計算ができる機能を実装する

※注：白色のトレイを使用する旨が仕様書に記載されていた

表 5-13 に指摘件数を集計した結果を示す．指摘件数の集計を行うにあたって，指摘を有効な指摘と無効な指摘に分類した．次の 4 つに該当する指摘は無効と判断した．(1)レビュー対象の製品の仕様から大きく逸脱する改善案，(2)誤字脱字・文章表現の分かりにくさなどの文章の記述に関する指摘，(3)仕様の内容に対するレビューアの認識が正しいかを確認するための単なる質問，(4)指摘ミス．(1)に該当する指摘は実際の製

品開発プロジェクトにおいても仕様の修正や再考に繋がらないことから，無効な指摘とした．(2)については，仕様の誤読に繋がる可能性があるため実際の製品開発プロジェクトにおいては修正することが望ましいが，あくまで文章表現の問題であって仕様そのものの内容に問題があるわけではないことから，集計の対象外とした．(3)については，仕様書を読んだレビューアが疑問を持つということは仕様書の文章表現に何らかの問題がある可能性はあるが，レビューアが理解した内容は正しかったため問題ないと考えた．(4)については，仕様書に記載されている内容に対して「記載がない」という指摘をした場合や，同じ内容の指摘を重複して導出していた場合には指摘ミスと考え，集計から除外した．同じ内容の指摘を重複して導出していた場合は 1 件として集計した．

表 5-13　実験結果の概要

	1 回目		2 回目	
	従来手法（PBR）	提案手法（ACA）	従来手法（PBR）	提案手法（ACA）
指摘件数　平均値（件）	17.9	24.4	10.7	22.0
中央値（件）	15.0	17.0	11.0	17.0
標準偏差	6.4	14.5	6.1	10.7
レビュー実施時間　平均値（分）	173.7	386.6	114.6	315.6
指摘 1 件の導出にかかった時間（分）	10.3	18.0	14.0	16.5

1 回目のレビューの指摘件数の平均値と中央値は PBR では 17.9 件と 15.0 件に対し提案手法では 24.4 件と 17.0 件，2 回目のレビューは PBR では 10.7 件と 11.0 件に対し提案手法では 22.0 件と 17.0 件であり，1 回目と 2 回目ともに PBR よりも提案手法の方が多く指摘を導出できた．PBR を使用した場合と提案手法を使用した場合それぞれの指摘件数の平均値差の検定を並べ替え検定により行ったところ，1 回目のレビューは $p = 0.339$，2 回目のレビューは $p = 0.042$ であり，2 回目のレビューでは 5%水準で指摘件数に有意差があった．1 回目のレビュー指摘件数の標準偏差は PBR では 6.4 に対し提案手法は 14.5，2 回目のレビューの PBR では 6.1 に対し提案手法は 10.7 であり，1 回目と 2 回目ともに PBR よりも提案手法のほうが大きく，被験者ごとの指摘件数のばらつきは提案手法のほうが大きかった．レビュー実施時間の平均値は，1 回目の PBR

では 173.7 分に対し提案手法では 386.6 分，2 回目の PBR では 114.6 分に対し提案手法では 315.6 分であり，PBR より提案手法のほうがレビューの実施に多くの時間を要した．1 件の指摘を導出するのにかかった時間を見ても，1 回目の PBR では 10.3 分に対し提案手法では 18.0 分，2 回目の PBR では 14.0 分に対し提案手法では 16.5 分であり，PBR より提案手法のほうが多くの時間を要した．提案手法は BBS やユースケースといった中間成果物を作成する必要があるため，中間成果物のない PBR よりもレビュー完了までに多くの時間がかかった．

PBR と提案手法の指摘件数の分布を図 5-9 に示す．1 回目のレビューと 2 回目のレビューともに，PBR よりも提案手法のほうが被験者ごとの指摘件数のばらつきは大きいが，指摘件数は多い傾向が見られた．1 回目のレビューよりも 2 回目のレビューのほうが，PBR と提案手法の指摘件数の差が大きかった．

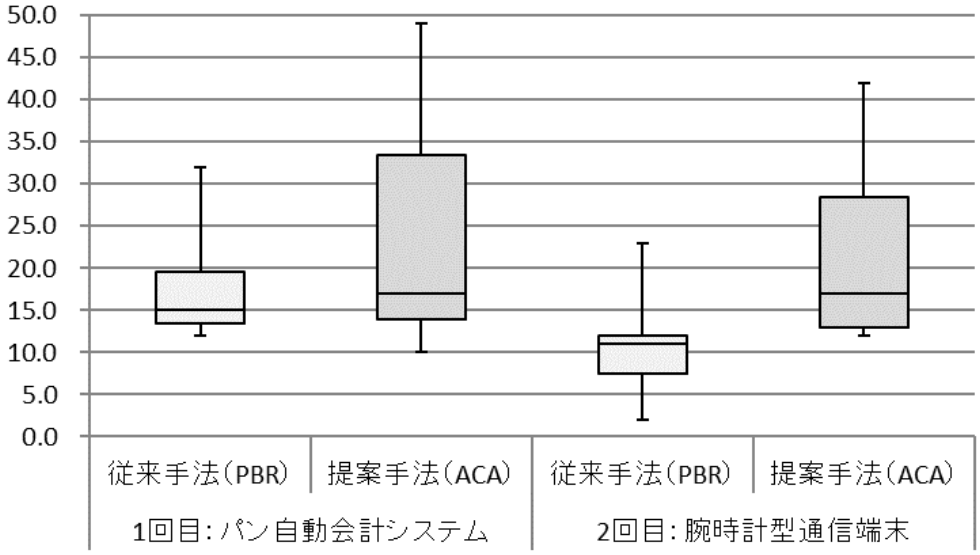


図 5-9　指摘件数の分布

被験者全員の指摘を重要度別に分類し，重要度別の指摘件数の平均値を集計した．重要度の分類は前節の実験と同様に表 5-3 を利用した．集計結果を表 5-14 に示す．重要度が「高」の指摘について，1 回目のレビューの PBR では 5.1 件に対し提案手法では 11.0 件，2 回目のレビューの PBR では 2.0 件に対し提案手法では 12.1 件であり，提案手法のほうが多く指摘を導出した．重要度が「中」の指摘について，1 回目のレビュ

ーの PBR では 4.4 件に対して提案手法では 9.6 件, 2 回目のレビューの PBR では 2.4 件に対し提案手法では 4.1 件であり, 提案手法のほうが多く指摘を導出した. 特に, 2 回目のレビューにおける重要度が「高」の指摘件数の平均は PBR の 2.0 件に対して提案手法では 12.1 件と約 6 倍であり, 大きく差があった. 重要度が「低」の指摘について, 1 回目のレビューの PBR では 8.3 件に対して提案手法では 3.9 件, 2 回目のレビューの PBR では 6.3 件に対し提案手法では 5.7 件であり, 2 回目のレビューでは差は小さいが, 1 回目と 2 回目ともに PBR のほうが多く指摘を導出した.

並べ替え検定により PBR と提案手法の指摘件数の平均値差の検定を行った結果, 1 回目のレビューでは重要度が「高」の指摘は $p = 0.880$, 重要度が「中」の指摘は $p = 0.101$, 重要度が「低」の指摘は $p = 0.010$ となり, 重要度が「低」の指摘件数は 1%水準で有意差があった. 2 回目のレビューでは重要度が「高」の指摘は $p = 0.001$, 重要度が「中」の指摘は $p = 0.374$, 重要度が「低」の指摘は $p = 0.851$ となり, 重要度が「高」の指摘件数は 1%水準で有意差があった.

表 5-14　重要度別の指摘件数の平均

重要度	1 回目			2 回目		
	従来手法（PBR）	提案手法（ACA）	*p* 値	従来手法（PBR）	提案手法（ACA）	*p* 値
高	5.1	11.0	0.880	2.0	12.1	0.001
中	4.4	9.6	0.101	2.4	4.1	0.374
低	8.3	3.9	0.010	6.3	5.7	0.851

※単位は「件」. *p* 値は並べ替え検定により算出.

重要度別の指摘件数の分布を図 5-10 に示す. 重要度が「高」と「中」の指摘は, 1 回目と 2 回目ともに PBR よりも提案手法のほうが指摘件数は多い傾向が見られた. 重要度が「低」の指摘は, 1 回目のレビューでは提案手法よりも PBR の方が指摘件数は多い傾向が見られ, 2 回目のレビューでは PBR と提案手法の指摘件数の差はほとんどなかった.

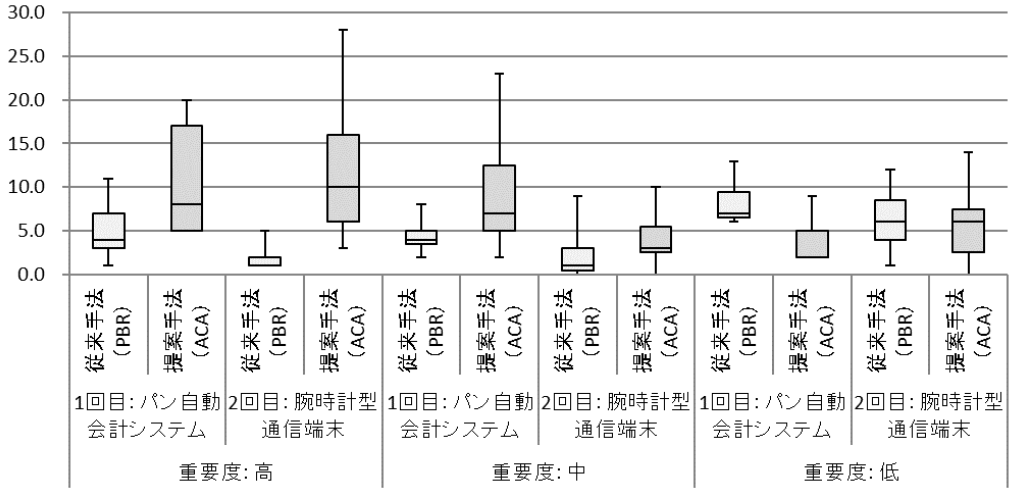


図 5-10　重要度別の指摘件数の分布

レビュー時にレビューアが考慮した範囲を確認するために，レビューアが導出した指摘の内容を分類し，各被験者が指摘できた分類の数を集計した．以降，指摘の内容の分類を指摘分類と呼ぶ．パン自動会計システムに対する指摘分類の一部を図 5-11 に，腕時計型通信端末に対する指摘分類の一部を図 5-12 に示す．BBS のフレームワークの構造をベースに，どの要素に対するどのような指摘かを抽象的に表現したものとなっている．

図 5-11　パン自動会計システムに対する指摘分類（一部）

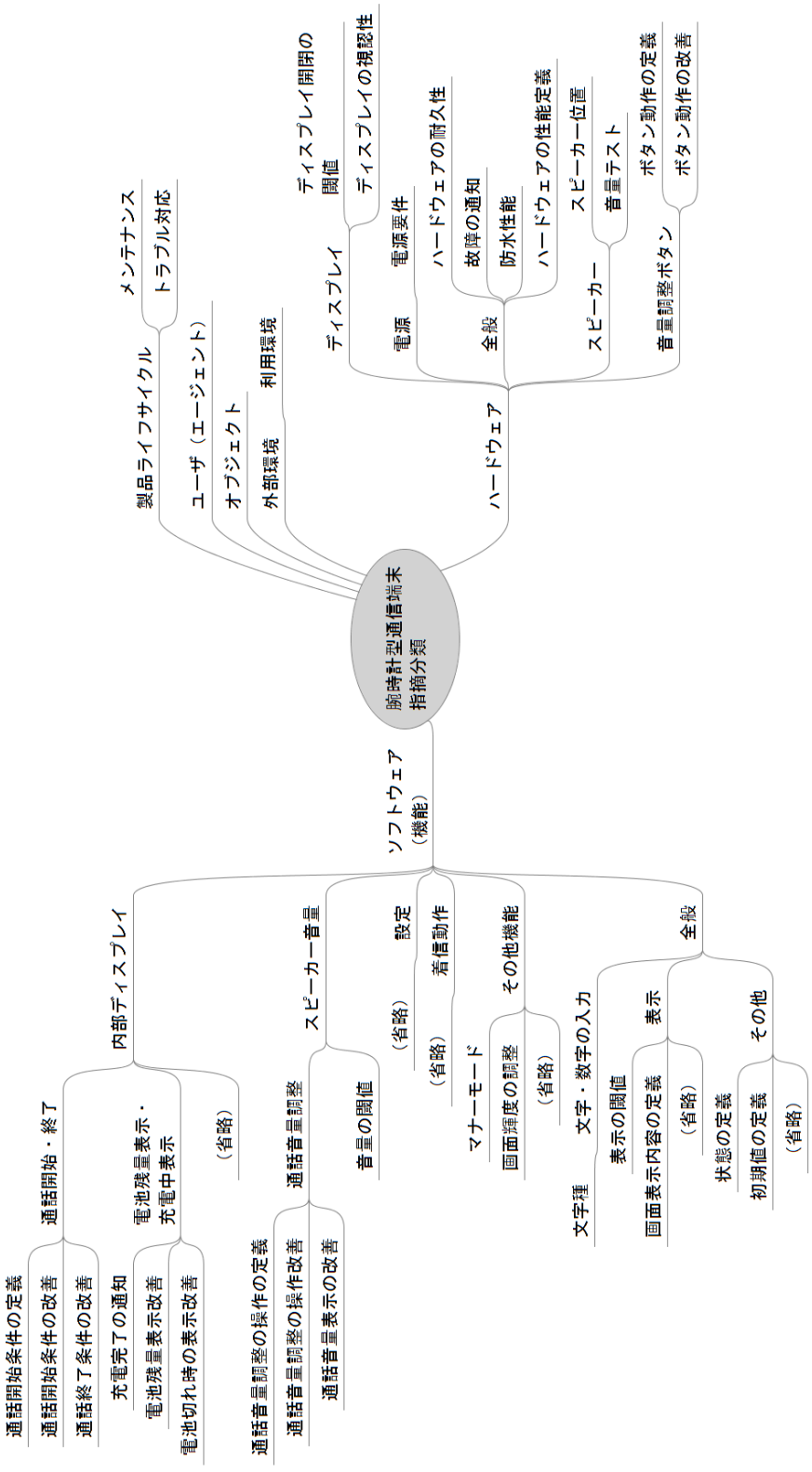


図 5-12　腕時計型通信端末に対する指摘分類（一部）

被験者全員の指摘を重要度別・指摘分類別に集計し，件数の平均値を算出した．重要度の定義はこれまでと同様に表 5-3 を利用した．集計結果を表 5-15 に示す．重要度が「高」の指摘について，1 回目のレビューの PBR では 4.4 件に対し提案手法では 7.1 件，2 回目のレビューの PBR では 1.9 件に対し提案手法では 8.0 件であり，提案手法のほうが多く指摘を導出した．重要度が「中」の指摘について，1 回目のレビューの PBR では 3.6 件に対して提案手法では 5.9 件，2 回目のレビューの PBR では 2.0 件に対し提案手法では 3.3 件であり，提案手法のほうが多く指摘を導出した．特に，2 回目のレビューにおける重要度が「高」の指摘件数の平均は PBR の 1.9 件に対して提案手法では 8.0 件と約 4 倍であり，大きく差があった．重要度が「低」の指摘について，1 回目のレビューの PBR では 6.1 件に対して提案手法では 3.4 件，2 回目のレビューの PBR では 4.9 件に対し提案手法では 3.9 件であり，1 回目と 2 回目ともに PBR のほうが多く指摘を導出した．

並べ替え検定により指摘分類の件数の平均値差の検定を行った結果，1 回目のレビューでは重要度が「高」の指摘は $p = 0.207$，重要度が「中」の指摘は $p = 0.165$，重要度が「低」の指摘は $p = 0.047$ となり，重要度が「低」の指摘件数は 5%水準で有意差があった．2 回目のレビューでは重要度が「高」の指摘は $p = 0.002$，重要度が「中」の指摘は $p = 0.354$，重要度が「低」の指摘は $p = 0.558$ となり，重要度が「高」の指摘件数は 1%水準で有意差があった．

表 5-15　重要度別の指摘分類の件数の平均

重要度	1 回目			2 回目		
	従来手法（PBR）	提案手法（ACA）	p 値	従来手法（PBR）	提案手法（ACA）	p 値
高	4.4	7.1	0.207	1.9	8.0	0.002
中	3.6	5.9	0.165	2.0	3.3	0.354
低	6.1	3.4	0.047	4.9	3.9	0.558

※単位は「件」．p 値は並べ替え検定により算出．

重要度別の指摘分類の件数の分布を図 5-13 に示す. 重要度が「高」と「中」の指摘は, 1 回目と 2 回目ともに PBR よりも提案手法のほうが指摘件数は多い傾向が見られた. 重要度が「低」の指摘は, 1 回目のレビューでは提案手法よりも PBR の方が指摘件数は多い傾向が見られた.

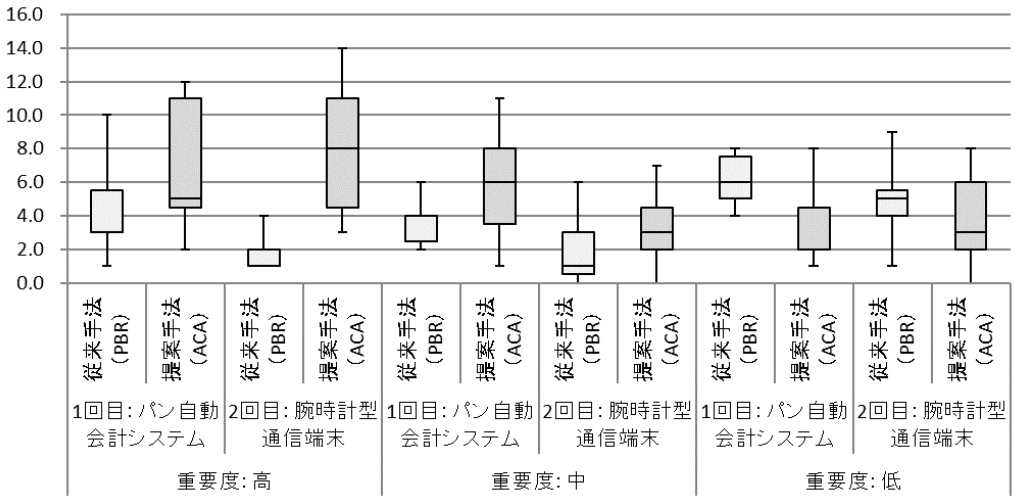


図 5-13　重要度別の指摘分類の件数の分布

被験者の人数ごとに指摘分類の網羅率を算出した結果を図 5-14 に示す. 図 5-14 は提案手法と PBR で導出したすべての指摘分類の総数を母数にした場合の指摘の網羅率を表す. i回目のレビューで提案手法を用いてj人で導出した指摘分類の件数の平均値をa_{ij}とし，PBR を用いてj人で導出した指摘分類の件数の平均値をp_{ij}とする. 被験者j人の組み合わせは${}_7C_j$通りあり，k番目の組み合わせの被験者らが導出した指摘分類の件数をa_{ijk}とp_{ijk}とする. i回目のレビューで提案手法を用いてj人で導出した指摘分類の網羅率をca_{ij}とし, PBR を用いてj人で導出した指摘分類の網羅率をcp_{ij}とする. i回目のレビューで提案手法を用いて導出したすべての指摘分類の集合をA_i，PBR を用いて導出したすべての指摘分類の集合をP_i，提案手法または PBR を用いて導出したすべての指摘分類の集合をT_iとする. $T_i = A_i \cup P_i$である.

a_{ij} と p_{ij} の算出式は以下のとおりである.

$$a_{ij} = \frac{\sum_{k=1}^{{}_7C_j} a_{ijk}}{{}_7C_j} \quad (1)$$

$$p_{ij} = \frac{\sum_{k=1}^{{}_7C_j} p_{ijk}}{{}_7C_j} \quad (2)$$

図 5-14 における網羅率の算出式は以下の通りである.

$$ca_{ij} = \frac{a_{ij}}{n(T_i)} \quad (3)$$

$$cp_{ij} = \frac{p_{ij}}{n(T_i)} \quad (4)$$

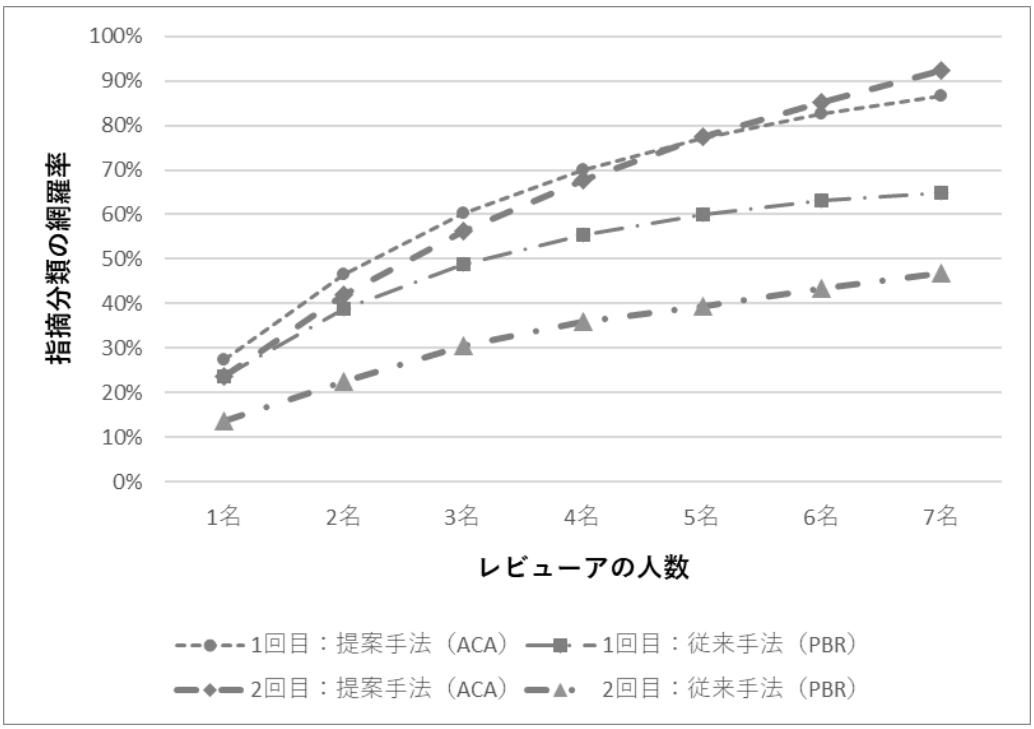


図 5-14　すべての指摘の分類を母数とした場合の網羅率

図 5-14 では, レビューアが 1 名の場合の網羅率は, 腕時計型通信端末のレビューにおいて PBR を使用した場合は約 10%, それ以外では 25%～30%となり, 腕時計型通信端末のレビューに PBR を使用した場合は 1 名での網羅率が低かった. どのレビュー技法・レビュー対象でもレビューアの人数が増えるごとに網羅率が増えた. 7 名が提案手法を使用した場合の網羅率は 85%～90%程度, PBR を使用した場合の網羅率は

約 65%・45%となり，7 名でレビューした場合の網羅率は PBR より提案手法のほうが高く，提案手法を使用した場合はレビュー対象に関わらずに同程度の網羅率を達成した．

BBS で被験者が導出した境界条件のうち，仕様書に記載のなかった境界条件を何名の被験者が導出できたか集計した結果を表 5-16 に示す．悪条件の発生源である「ハードウェア」・「オブジェクト」・「外部環境」・「ユースケース」に該当する境界条件を対象とした．各悪条件の発生源に該当する境界条件のうち，仕様書に記載のなかった境界条件に対して，当該境界条件を BBS で導出できた人数を集計し，悪条件の発生源ごとに平均値を算出した．提案手法の訓練で使用した BBS のサンプルに記載されていた境界条件と重複するものは，レビューア個人の気付きによって導出されたものではないと考えられるため集計から除外した．レビューアが導出した境界条件のうち，文字列が完全に一致していなくても同一の境界条件を表すと考えられるものは，筆者の判断で同一のものとして集計した．例えば，「お年寄り」と「老人」や，「電波状況」と「電波環境」は同一のものとした．1 回目のレビューと 2 回目のレビューともに，エージェント・オブジェクト・外部環境に該当する境界条件で仕様書に記載のないものを導出したレビューアの人数の平均値は 1～2 人の間だった．ハードウェアに関する境界条件は仕様書に記載されていたもののみ BBS で導出されていたことから，表 5-16 のハードウェアに該当する部分は「-」としている．

表 5-16　仕様書に記載のない境界条件を導出したレビューアの人数の平均

悪条件の発生源	1 回目	2 回目
エージェント	1.2 人	1.4 人
ハードウェア	-	-
オブジェクト	1.3 人	1.3 人
外部環境	1.2 人	1.6 人

※ハードウェアの境界条件については仕様書に記載のないものは導出されなかった

レビューの価値を定量化するメトリクスとして，「修正規模の金額効果」が提案されている[96]．これは，レビューの指摘内容から予測される修正規模を金額に換算したメトリクスである．修正規模の金額効果は，「修正範囲に関するポイント」，「発見工程に関す

るポイント」,「単位ポイントあたりの修正工数」,「時間単価」の積により算出できる. 修正範囲に関するポイントは表 5-17, 発見工程に関するポイントは表 5-18 の定義に従う. 文献[96]では修正範囲に関するポイントは, 欠陥を修正する際の影響範囲によりランク分けされている. 本研究では指摘の分類として表 5-3 の重要度を用いており, 重要度が高いほど仕様作成時に考慮していなかった境界条件を仕様に取り込む必要があることから, 修正する際の影響範囲が広くなることが想定される. そこで, 重要度の「高」「中」「低」が修正範囲のランクの「大」「中」「小」に該当するものとした. 発見工程に関するポイントは表 5-18 に示した通り工程ごとに定義される. 本実験は要求定義段階における要求仕様書のレビューを対象としたため, 発見工程のポイント数は「5」とした. 文献[96]での試算に基づき, 単位ポイントあたりの修正工数は 4 時間, 時間単価は 5,000 円とした.

表 5-17　修正範囲とポイント

文献[96]における修正範囲のランク	本研究における指摘の重要度	ポイント
大	高	5
中	中	2
小	低	1

※文献[96]の表 3 を元に筆者が作成した.

表 5-18　発見工程とポイント

工程	ポイント
要求定義	5
詳細設計	2
コーディング	1

※文献[96]の表 4 を引用.

表 5-14 に示した重要度別の指摘件数の平均値をもとに，修正規模の金額効果を算出した．レビュー所要コストは，表 5-13 のレビュー実施時間の平均値と時間単価の積から算出した．修正規模の金額効果は，仕様のレビューで欠陥を検出したことによって後工程で発生しなくなった費用であり，プロジェクトで支払わなくて済んだ費用を表す．レビュー所要コストは，レビューという活動そのものを実施するためにかかった費用であり，プロジェクトでレビューを実施するために追加で支払った費用を表す．これらの費用の差は，どれだけの費用をプロジェクト全体で節約できたかを表す．そこで本研究では，修正規模の金額効果とレビュー所要コストの差をレビューの費用対効果と考えた．レビューの費用対効果の算出結果を表 5-19 および図 5-15 に示す．PBR と比較すると提案手法のほうがレビュー所要コストは大きいが，費用対効果も大きかった．

表 5-19　レビューの費用対効果

	1 回目		2 回目	
	従来手法（PBR）	提案手法（ACA）	従来手法（PBR）	提案手法（ACA）
修正規模の金額効果	¥4,285,714	¥7,800,000	¥2,114,286	¥7,471,429
レビュー所要コスト	¥14,476	¥32,214	¥9,548	¥26,298
費用対効果	¥4,271,238	¥7,767,786	¥2,104,738	¥7,445,131

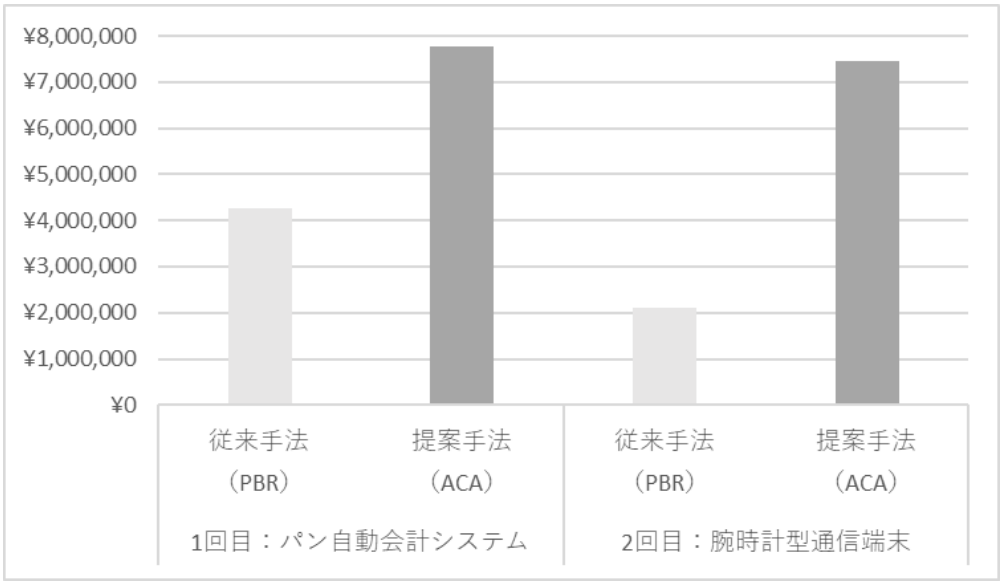


図 5-15　レビューの費用対効果

(エ)アンケートの集計結果

表 5-20 および図 5-16 に, レビュー時の考慮範囲に関するアンケートの集計結果を示す. アンケートの回答結果を次の通りに点数に置き換え平均値を算出した.「明確に考慮した＝5 点」,「なんとなく考慮した＝4 点」,「どちらとも言えない＝3 点」,「あまり考慮しなかった＝2 点」,「まったく考慮しなかった＝1 点」.

1 回目のレビューについて, 利用者/顧客の視点は PBR の 4.4 に対し提案手法は 4.1, 設計者/開発者の視点は PBR の 4.4 に対し提案手法は 3.6, テスト担当者の視点は PBR の 4.3 に対し提案手法は 4.3, ハードウェアの悪条件は PBR の 2.1 に対し提案手法は 3.6, オブジェクトの悪条件は PBR の 3.0 に対し提案手法は 3.4, 外部環境の悪条件は PBR の 2.9 に対し提案手法は 3.6, ユースケースの悪条件は PBR の 2.9 に対し提案手法は 4.1 となり, すべての悪条件について PBR に比べ提案手法のほうが点数が高かった. 2 回目のレビューについて, 利用者/顧客の視点は PBR の 4.9 に対し提案手法は 4.3, 設計者/開発者の視点は PBR の 4.3 に対し提案手法は 3.3, テスト担当者の視点は PBR の 4.4 に対し提案手法は 2.9, ハードウェアの悪条件は PBR の 1.1 に対し提案手法は 4.4, オブジェクトの悪条件は PBR の 1.3 に対し提案手法は 3.6, 外部環境の悪条件は PBR の 1.7 に対し提案手法は 4.4, ユースケースの悪条件は PBR の 2.3 に対し提案手法は 4.4 となり, すべての悪条件について PBR に比べ提案手法のほうが点

数が高かった. 1 回目のレビューに比べ 2 回目のレビューのほうが PBR と提案手法の点数の差が大きかった. 1 回目のレビュー・2 回目のレビューともに, PBR のレビューの視点は PBR と提案手法で点数の差が小さかったが, 悪条件は PBR と提案手法で点数の差が大きかった.

並べ替え検定によりアンケートの得点の平均値差の検定を行った結果, 1 回目のレビューについて, 利用者/顧客の視点は $p = 0.559$, 設計者/開発者の視点は $p = 0.073$, テスト担当者の視点は $p = 0.073$, ハードウェアの悪条件は $p = 0.066$, オブジェクトの悪条件は $p = 0.674$, 外部環境の悪条件は $p = 0.442$, ユースケースの悪条件は $p = 0.084$ となり, いずれのレビューの視点も考慮度合は 5%水準で有意差はなかった. 2 回目のレビューについて, 利用者/顧客の視点は $p = 0.103$, 設計者/開発者の視点は $p = 0.152$, テスト担当者の視点は $p = 0.029$, ハードウェアの悪条件は $p = 0.001$, オブジェクトの悪条件は $p = 0.002$, 外部環境の悪条件は $p = 0.003$, ユースケースの悪条件は $p = 0.009$ となり, すべての悪条件の視点について考慮度合は 1%水準で有意差があった.

表 5-20　レビュー時の考慮範囲に関するアンケート結果

レビューの視点	1 回目			2 回目		
	従来手法（PBR）	提案手法（ACA）	*p* 値	従来手法（PBR）	提案手法（ACA）	*p* 値
利用者/顧客の視点	4.4	4.1	0.559	4.9	4.3	0.103
設計者/開発者の視点	4.4	3.6	0.073	4.3	3.3	0.152
テスト担当者の視点	4.3	4.3	0.073	4.4	2.9	0.029
ハードウェアの悪条件	2.1	3.6	0.066	1.1	4.4	0.001
オブジェクトの悪条件	3.0	3.4	0.674	1.3	3.6	0.002
外部環境の悪条件	2.9	3.6	0.442	1.7	4.4	0.003
ユースケースの悪条件	2.9	4.1	0.084	2.3	4.4	0.009

※*p* 値は並べ替え検定により算出.

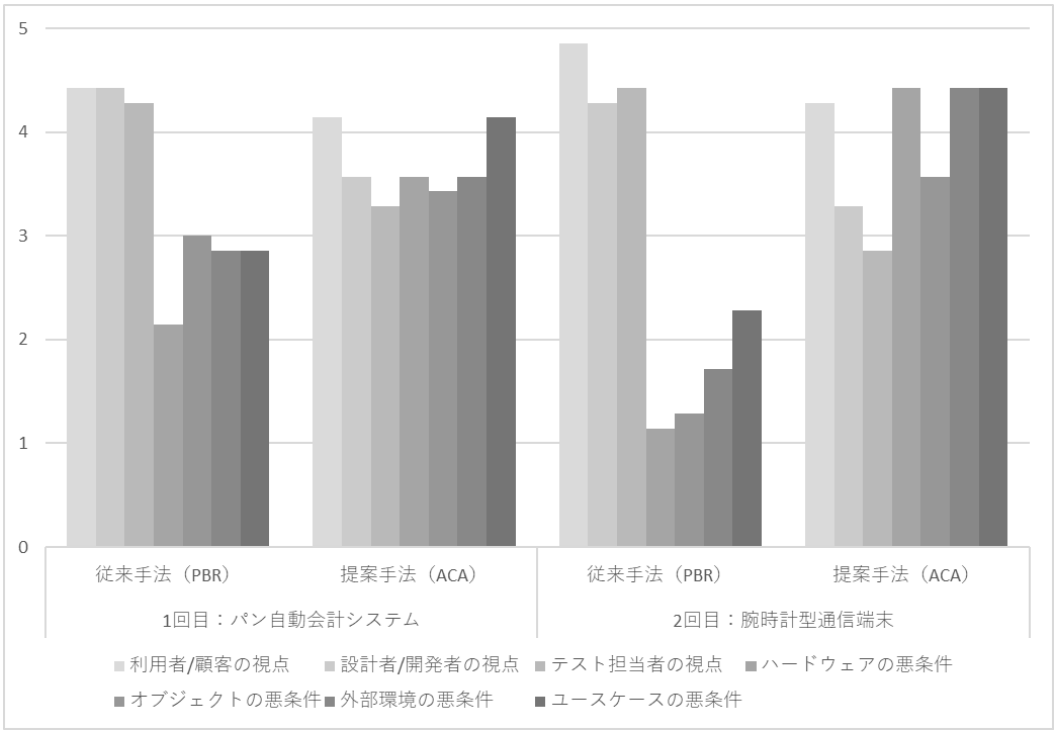


図 5-16　レビュー時の考慮範囲に関するアンケート結果

レビュー技法の実務での活用可能性に関するアンケートの結果を表 5-21 に示す. 表 5-21 では, アンケートの回答選択肢ごとに回答を選択した人数を集計し, 1 回目のレビュー後のアンケートと 2 回目のレビュー後のアンケートの回答者数を合計した. 回答者数は PBR と提案手法で同数となり,「非常にそう思う」を選択した被験者が 1 名,「そう思う」を選択した被験者が 9 名,「どちらとも言えない」を選択した被験者が 4 名だった. 「思わない」と「まったく思わない」を選択した被験者はいなかった.

表 5-21　レビュー技法の実務での活用可能性に関するアンケート結果

回答選択肢	従来手法（PBR）	提案手法（ACA）
非常にそう思う	1 人	1 人
そう思う	9 人	9 人
どちらとも言えない	4 人	4 人
思わない	0 人	0 人
まったく思わない	0 人	0 人

レビュー技法を実務で使いたいと思う理由を問うアンケートの結果を図 5-17 に，使いたいと思わない理由を問うアンケートの結果を図 5-18 に示す．アンケートの回答選択肢で「その他」を選んだ被験者による自由記述の回答を表 5-22 に示す．

図 5-17 より提案手法を実務で活用したいと思う理由は，「指摘の途中経過を含めて顧客に説明しやすそうだから」を選択した被験者がのべ 6 名でもっとも多く，PBR では「指摘の漏れを減らせそうだから」を選択した被験者がのべ 6 名でもっとも多かった．次に多かったのは，提案手法では「普段思いつかない指摘を思いつくことができそうだから」・「指摘の漏れを減らせそうだから」を選択した被験者が各のべ 4 名，「重要な指摘を出せそうだから」を選択した被験者がのべ 3 名だった．PBR では「普段思いつかない指摘を思いつくことができそうだから」・「重要な指摘を出せそうだから」を選択した被験者が各のべ 4 名，「指摘の途中経過を含めて顧客に説明しやすそうだから」を選択した被験者がのべ 3 名だった．PBR では「レビューの時間を短縮できそうだから」を選択した被験者がのべ 1 名いたが，提案手法に対してこの選択肢を選んだ被験者はいなかった．

図 5-18 より提案手法を実務で活用できるか分からない理由は，「実施するのに時間がかかるから」・「技法を理解できていないから」を選択した被験者が各のべ 2 名で最も多く，次いで「難易度が高いから」・「効果があるか不明だから」を選択した被験者が各のべ 1 名だった．PBR では「難易度が高いから」を選択した被験者がのべ 2 名で最も多く，次いで「実施するのに時間がかかるから」・「技法を理解できていないから」・「効果があるか不明だから」・「自分の担当する製品にこの技法を適用するイメージがわかないから」を選択した被験者が各のべ 1 名だった．

表 5-22 より，提案手法に対する意見として，考慮範囲が広がりすぎることに対する懸念と提案手法の使い方の理解不足に関する意見があった．PBR に対する意見として，レビューの標準化に繋がるという意見やレビュー観点の活用に対する前向きな意見があった一方，同様のレビュー観点ですでにレビューを実施しているという意見や，習熟の難しさに対する意見があった．

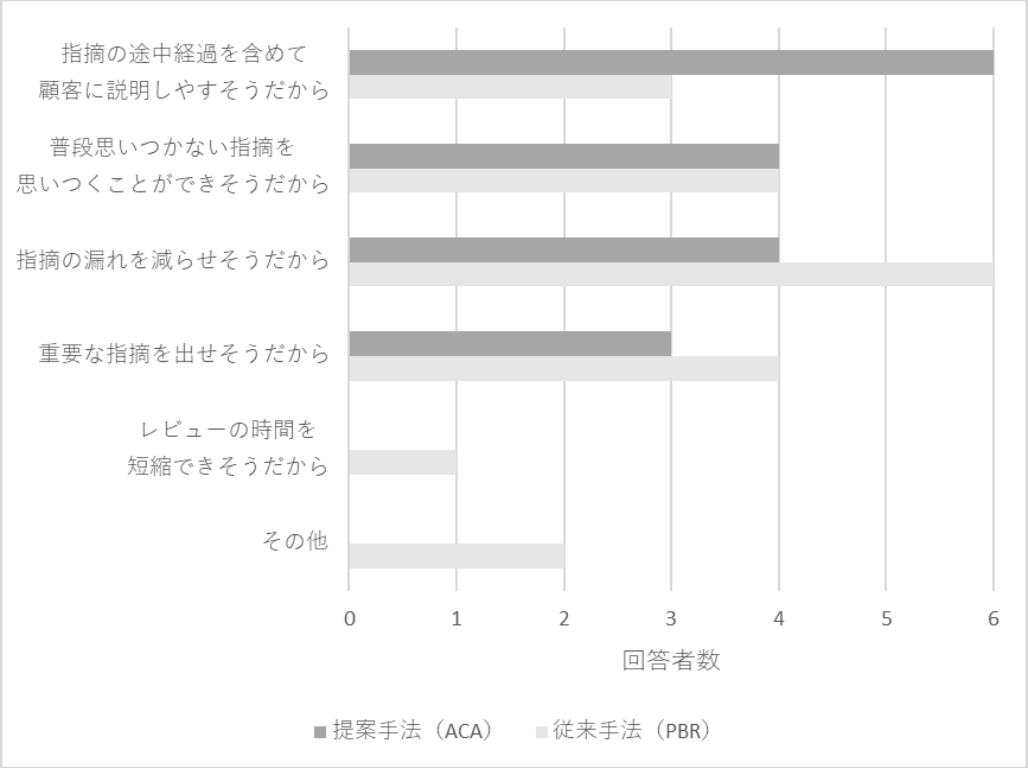


図 5-17　レビュー技法を実務で活用したいと思う理由

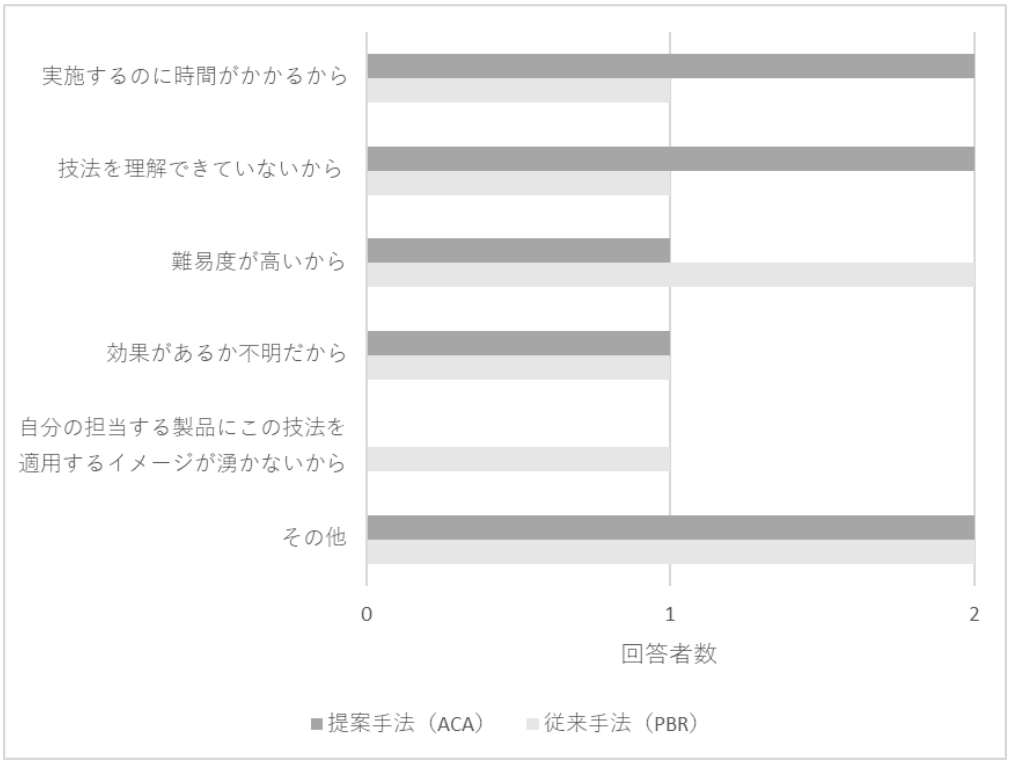


図 5-18　レビュー技法を実務で活用できるか分からない理由

表 5-22　レビュー技法の実務での活用可能性に関する自由記述

レビュー技法	回答
従来手法（PBR）	・ レビューの標準化をするうえで指標に使えそう.
	・ 観点は異なるが，観点を決めてそれにそってレビューを行う手法は利用している. 今回使用した観点も取り入れられないか検討しようと思う.
	・ レビュー観点という意味ではすでに実施している.
	・ 習熟には各立場の経験が必要だと思った.
提案手法（ACA）	・ 影響する範囲が多大すぎて作成が難しくなりそう.
	・ もう少し理解して使いたい.

(オ)考察

本実験の目的に記載した 3 つの確認事項に対して，実験結果に基づき議論する. まず「(1) 要求仕様書の記述の詳細さに関わらず，提案手法によりレビューアの考慮範囲が広くなり，製品の改善に繋がる重要度の高い指摘を従来のレビュー技法より多く導出できること」について，重要度別の指摘件数，指摘分類数，指摘分類の網羅率，悪条件の考慮度合いに関するアンケート結果に基づいて考察する.

表 5-13 および図 5-9 から，提案手法を利用した場合のほうが PBR を使用した場合よりも指摘件数は多い傾向があった. パン自動会計システムのレビューよりも腕時計型通信端末のレビューのほうが，PBR と提案手法の指摘件数の差が顕著である. 境界条件の整理のプロセスが明示的でない PBR ではレビュー結果が要求仕様書の記述の詳細度の影響を受けやすいが，提案手法ではフレームワークを活用して明示的に境界条件を整理するため，要求仕様書の記述の詳細度による影響が少なかったと考える.

表 5-14 および図 5-10 から，重要度が「高」「中」の指摘は提案手法を使用した場合のほうが PBR を使用した場合よりも指摘が多い傾向があった. 腕時計型通信端末のレビューにおける重要度「高」の指摘件数は 1%水準で有意差があり，要求仕様書の記述が詳細でない場合，提案手法を利用することで，想定外の境界条件を考慮した仕様の改善案を多く導出できることが確認された. 重要度「低」の指摘は PBR を使用した場合のほうが指摘件数が多い傾向が見られた. パン自動会計システムのレビューにおける

重要度「低」の指摘件数は 1%水準で有意差があり，要求仕様書の記述が詳細な場合に記述された境界条件の範囲内で問題点を検出するには，仕様書の記述に対して直接レビュー観点を照らし合わせて問題点を見つけるという従来のレビュー技法のアプローチのほうが効果を出しやすいと考えられる．表 5-15 および図 5-13 からも，同様の傾向が確認でき，指摘分類の件数としても，PBR より提案手法のほうが重要度「高」「中」の指摘分類は多く，提案手法より PBR のほうが重要度「低」の指摘分類は少なかった．

図 5-14 に示した指摘の網羅率は，提案手法ではパン自動会計システムのレビューでも腕時計型通信端末のレビューでも同等の網羅率を達成しており，要求仕様書の記述の詳細さに関わらず指摘を導出できている．どちらのレビュー対象でも PBR に比べ，提案手法のほうが人数が増えるごとに網羅率の増加が大きかった．PBR よりも提案手法のほうがレビューアが広範囲の境界条件を考慮して指摘できたことで，網羅率の増加が大きくなったと考えられる．PBR では腕時計型通信端末の場合の網羅率が低く，要求仕様書の記述の詳細さに影響を受けたと考えられる．

表 5-16 から，提案手法を使用した場合に仕様書に記載のない境界条件を導出できたレビューアの平均人数は 1 人～2 人の間であり，提案手法によりレビューアの考慮範囲が広がった結果として，仕様書に記載のない境界条件の導出可否はレビューア個々人の気付きへの依存が大きくなったと考えられる．図 5-14 で，提案手法を使用した場合にレビューアの人数が増えるごとの網羅率の増加が大きかったのは，仕様書に記載のない境界条件の導出結果がレビューアごとに異なることが影響していると考えられる．

表 5-20 および図 5-16 のアンケート結果から，腕時計型通信端末のレビューで PBR を使用した場合は悪条件の考慮度合いの数値が特に低く，1.1～2.3 となっている．提案手法を使用した場合はパン自動会計システムのレビューでも腕時計型通信端末のレビューでも，悪条件の考慮度合いは 3.4～4.4 と高い数値になっている．提案手法のフレームワークとガイドワードを利用することで，要求仕様書の記述が詳細でない場合でも，レビューアが悪条件を考慮できたことがわかる．PBR の視点である「利用者/顧客の視点」・「設計者/開発者の視点」・「テスト担当者の視点」はパン自動会計システムのレビューと腕時計型通信端末のレビューともに，どちらのレビュー技法を使用した場合でも約 3～4 点台の数値となっている．PBR の視点は，レビュー実施時に特別に視点を提示しなくても，実務に携わるエンジニアはある程度考慮可能であると考えられる．PBR の視点は提案手法を利用した場合でも考慮度合いが大きく下がっていないこと，悪条

件の視点は提案手法を利用した場合に考慮度合いが増加していることから，提案手法を利用した場合のほうが PBR よりもレビューアが考慮できる範囲が広がっていると言える．

以上から，「(1) 要求仕様書の記述の詳細さに関わらず，提案手法によりレビューアの考慮範囲が広くなり，製品の改善に繋がる重要度の高い指摘を従来のレビュー技法より多く導出できること」が確認できたと言える．

次に「(2) 費用対効果が従来のレビュー技法と比較して高いか」について，費用対効果の算出結果に基づいて考察する．表 5-13 から提案手法のレビューにかかる時間は PBR の 2〜3 倍程度に増加しているが，費用対効果は表 5-19 および図 5-15 から提案手法の方が大きいことが分かる．上流工程で指摘を導出するほど費用対効果は高くなるが，上流工程であるほど境界条件が不明確な場合が多く従来手法のアプローチでは指摘を導出できる範囲が限定される．提案手法は仕様書の記述があいまいな場合でも想定外の境界条件を考慮し重要度の高い指摘を導出できるため，大きな費用対効果に繋がったと考えられる．

最後に「(3) 製品検証の実務に携わるエンジニアが提案手法の効果を認め実務で活用したいと思うか」について，レビュー技法の実務での利用可能性に関するアンケート結果およびインタビューに基づき考察する．表 5-21 のアンケート結果から，レビュー技法の実務での活用可能性は PBR と提案手法で差が無かった．レビュー技法を実務で活用したいと思うかという問いに対して，14 名中 10 名の被験者が「非常にそう思う」または「そう思う」と回答したことから，実務において提案手法または PBR を利用する価値が，実務に携わるエンジニアの主観的評価において認められたと言える．

被験者が提案手法を実務で活用したいと考えた理由のうち最も多かったのは，図 5-17 から「指摘の途中経過を含めて顧客に説明しやすそうだから」という理由だった．提案手法に対してこの理由を選択した被験者は PBR に対してよりも多かった．提案手法を使用する場合，BBS などの中間成果物を作成することによって，レビューアが指摘を導出する際の思考の途中経過が可視化されるため，レビューの結果を他者に対して説明しやすくなるという点が被験者から評価されたと考えられる．レビューアの思考過程を可視化できることは，設計者や他のレビューアによるレビューの妥当性確認やレビュー結果の振り返り，および指摘の採用有無を議論し合意形成する際に有用である．経験の浅いレビューアがベテランのレビューアの思考過程を学習することが可能となり，教

育面の効果も期待できる.

被験者がPBRを実務で活用したいと考えた理由のうち最も多かったのは，図5-17から「指摘の漏れを減らせそうだから」という理由だった．提案手法に対してこの理由を選択した被験者は PBR に対してよりも少なかった．提案手法を使用する場合，境界条件を複数の視点から分析できるが，最終的に指摘を導出するための視点は悪条件に固定される．PBR は様々なステークホルダーの視点で網羅的に指摘を導出できること，所与のレビュー観点を網羅的にチェックすることから，レビューアは網羅的にレビューを実施できたと感じやすく，「指摘の漏れを減らせそうだ」という被験者の評価に繋がったと考えられる．提案手法による境界条件の整理と悪条件の検討のアプローチは従来のレビュー技法には存在しないため，そのアプローチの効果がレビューアにとって理解しづらいという課題が，提案手法には存在すると考えられる．境界条件の整理と悪条件の検討の結果には正解がないため，レビューアはどこまで考えれば網羅したと言えるか分からず，考慮漏れがないか不安に感じ，「指摘の漏れを減らせそうだから」と回答した被験者が少なかったと考えられる．しかしながら，提案手法を業務で活用したい理由として「指摘の漏れを減らせそうだから」と回答した被験者は 4 名いたことから，指摘の漏れを減らす効果について，被験者から一定の評価はされていると言える．図 5-14 から提案手法は PBR よりも指摘の網羅度合いが高いため，具体的な効果の数値も含めて，レビューアが提案手法の効果を理解しやすいような教育プログラムを今後整備していく必要がある.

図 5-18 から各レビュー技法を実務で活用できるか分からない理由として，実施するのにかかる時間の問題，レビュー技法に関する理解度や実施の難易度の問題，効果に対する疑問が，どのレビュー技法でも被験者から選択された．表 5-22 においても「習熟には各立場の経験が必要だと思った」・「影響する範囲が多大すぎて作成が難しくなりそう」・「もう少し理解して使いたい」といった，レビュー技法の理解・習熟や，実施の難易度に関する意見が出た．アドホックレビューと異なり，PBR や提案手法は決まったプロセスやレビュー観点やフレームワークを持っているため，これらの理解が必要となる点で，レビュー技法の理解度や実施の難易度の問題が挙げられたと考えられる．しかし，プロセスやフレームワークが体系化されていることで教育が可能になるととらえることもできる．文献[25]ではアドホックレビューと比べた PBR の優位な点として，体系的なプロセスが定義され教育が整備されている点を挙げている．体系的なプロセスが定義され

ている点では提案手法も PBR と同等である. レビューを実施するのにかかる時間の問題は, 既定のプロセスにそってレビューを行うことや複数のレビュー観点を使ってレビューを行うことで, アドホックなレビューよりも時間がかかることが実務上の懸念として挙げられたと考えられる. レビューで有効な指摘を導出するには, レビュー技法のプロセスやフレームワークの教育や学習の時間を取り, それらを利用して時間をかけてレビューを行う必要がある. 一方で, 実務においては, 重要な箇所や重大な欠陥に絞って短時間で効率的に指摘を検出したいというニーズは考えられ, 従来のレビュー技法のうち DBR や UBR のようにレビューの範囲を絞り込むアプローチが有効な場合もある. 各プロジェクトにおいてどれだけの時間をレビューにかけられるかによって, 適切なレビュー技法の利用を検討する必要がある.

実験が終了した後, 被験者に提案手法についてインタビューした結果, 次の意見が得られた.

- 提案手法は使いこなすのが難しいが, 使いこなすことができれば有効な武器になるだろう
- 実験が終わってから, 実務で行っているテストに, 異常状態に対する対策が立てられているかをチェックするという視点を組み込んだら, メンバーの視野を広げることができた
- 悪条件という製品にとってのマイナス面から指摘を考えるだけでなく, 製品の価値やユーザにとって良いことなどのプラス面から指摘を考えることも必要ではないか

悪条件を考慮して指摘を導出することが有用だという意見があった一方で, 悪条件を考慮するだけではレビューの視点としては不十分ではないかという意見もあった. 文献[36]では従来のレビュー技法の効果に関する既往の研究の文献レビューを行い, 「レビューの効果を高めるためには, 利用者が適材適所でレビュー技法を選び適用することが求められる」と結論付けている. 各レビュー技法のアプローチの違いおよび導出しやすい指摘を把握し, 適切に使い分ける必要がある. 使い分けの一つの方法として, 要求仕様作成の早い段階で提案手法を使用して考慮すべき境界条件を明確にし, 悪条件や必要な機能を洗い出したうえで, 要求仕様が詳細化された段階で PBR を使って各機能の詳細な動作条件を確認していくといった方法が考えられる. 複数のレビュー技法を適切に組み合わせる方法を検討することは, 今後の研究課題である.

以上から，実務における提案手法の活用の可能性は PBR と同等と考えられ，「(3) 製品検証の実務に携わるエンジニアが提案手法の効果を認め実務で活用したいと思うか」が確認できたと言える．

提案手法を適用可能な製品やシステムの範囲について，PBR などの従来手法と同様に，提案手法は特定の製品・システム・組織・開発プロセスに特化した方法ではないため，様々な組織や製品開発プロジェクトで適用可能と考える．大規模なシステムや，欠陥が致命的な問題に繋がるシステムに対しても適用可能と考えられるが，本研究のケーススタディおよび実験で扱った事例以外への適用の検証はできておらず，実際に提案手法を適用した際の効果の検証は今後の課題である．規模が大きいレビュー対象に提案手法を適用する場合，現実的な時間内でレビューを完了させるために，レビュー対象の分割や重要な箇所の特定，複数のレビューアでの作業の分担といった工夫が必要になる．

レビュー対象の要求仕様書の分量が多い場合， BBS の作成に多くの時間を要すことや，BBS が巨大になり境界条件の全体像を把握するのが困難になることが予想される．そのような場合はレビュー対象を機能やユースケースごとに分割し，優先順位をつけ，優先順位の高い部分に提案手法を適用することで，現実的な時間内でレビューを実施できる．リスクベースドテストでは，リスクの発生頻度と発生時の影響度を考慮してテストすべき機能の優先順位を決定する方法が提案されており[99]，同様の方法が提案手法でも利用可能である．分割した機能やユースケースごとに異なるレビューアを割り当て，複数名のレビューアが分担してレビューを行うこともできる．

欠陥が致命的な問題に繋がる製品やシステムがレビュー対象の場合，見逃してはならない欠陥が明らかであれば，当該の欠陥情報に基づく Defect based-reading を実施し欠陥の検出を行うとともに，提案手法を利用してそれ以外の問題も指摘するのが効果的と考えられる．前述の通り，複数のレビュー技法を適切に組み合わせる方法を検討することは，今後の研究課題である．

本実験の結果から，境界条件が明確な仕様書よりも境界条件があいまいな仕様書に対してのほうが，提案手法の効果が大きいことが確認された．境界条件が定まりきっていない要求分析や要求仕様書作成の段階から BBS による境界条件の構造化や悪条件の検討といった提案手法のプロセスを適用することで，要求仕様書の完成度を上げられるだけでなく，要求仕様書作成後のレビュー工数やテスト工数の削減が期待できる．

BBS により境界条件を構造化・可視化することで，要求仕様の設計者が考慮した境界条件が明確になり，境界条件の考慮不足の有無を第三者がチェックしやすくなるため，レビュー工数の削減に繋がると考えられる．要求仕様書作成の段階で BBS により境界条件が可視化されることで，テスト項目を作成する際にテスト担当者と設計者の間での要求仕様書の不明点に対する質問のやり取りが減ること，BBS をテスト分析で利用すればテスト分析の工程で改めて境界条件を整理する必要がなくなることから，テスト項目作成の時間が減る効果も想定される．提案手法を要求仕様書作成の段階から活用することによる，レビュー工数やテスト工数削減の効果について，今後検証を行う必要がある．

仕様書に記載のない境界条件を導出する方法や，複数の条件が組み合わさることで問題となるような悪条件は導出の難易度が高いと考えられるが，これらを導出する方法は提案手法において規定がなく，レビューアの気付きに依存する．前者は BBS のフレームワークを利用して境界条件をツリー構造で整理する際にレビューアの気付きにより補うこととしているが，気付かなかったものは導出されない．後者は悪条件を洗い出したうえで組み合わせを検討することは可能だが，組み合わせ数が多くなりすぎて，現実的な時間内で検討しきるのが難しいという懸念がある．これらの課題に対する解決策として，N-Fold Inspection を適用して複数のレビューアでレビューすることで気付きを増やす方法や，悪条件の組み合わせごとに複数のレビューアが分担してレビューを行う方法が考えられる．レビューアの人数を増やすとその分コストがかかることが難点ではあるが，図 5-14 の結果からも分かるようにレビューアの人数を増やすことは指摘の網羅度を上げるには確実な方法である．別の視点での解決策として，レビューアを増やす代わりに，過去のレビューで導出された境界条件や悪条件のデータを蓄積し，データに基づき境界条件や悪条件をサジェストする機能を有したレビュー支援ソフトウェアを開発することが考えられる．過去のプロジェクトのレビューで検出された境界条件と悪条件に加え，レビューで見逃されて後工程で重要な問題に繋がった境界条件や悪条件をデータとして蓄積し自然言語処理や機械学習等の技術を応用することで，レビューアが境界条件や悪条件を導出する際に関連する境界条件をサジェストすることや，問題を起こしやすい悪条件の組み合わせをサジェストすることが可能になる．過去のプロジェクトのデータに基づきレビュー支援ソフトウェアによるサジェストができれば，仕様書に記載されていない境界条件にレビューアが気付くことや，レビューア個人の経験から

は気付かなかった悪条件を導出することが可能となり，複数のレビューアがレビューを行うのと同様の効果を得ることができる．

(カ)実験の妥当性

本実験で被験者としたエンジニアは，レビュー対象の製品を実際に開発している企業のエンジニアではなく，製品開発企業から検証業務を請け負う立場の検証専門企業のエンジニアであるため，本実験の状況は，製品開発企業内で当該製品のエキスパートが要求仕様書をレビューする状況とは異なっている．しかしながら，検証専門企業のエンジニアが行う業務では，専門知識を持たない製品の検証や仕様書の欠陥検出を実施しなければならない状況があり，本実験の状況は検証専門企業に勤めるエンジニアにとっては実務として体験しうる状況を再現していると言える．製品開発企業においても，世の中にない革新的な製品の開発プロジェクト，当該企業が初めて開発する製品の開発プロジェクト，経験の浅い担当者がアサインされた製品開発プロジェクトでは同様の状況が発生しうる．

グループ 1 とグループ 2 は同一の被験者群ではないため，グループ 1 とグループ 2 の欠陥検出の能力が同一である保証はなく，各グループの被験者の能力差が本実験の結果に影響を及ぼした可能性がある．しかしながら，14 名の被験者をグループに分ける際にランダムに各グループに割り付けたことと，実際に割り付けられた各グループの検証業務経験の平均年数は表 5-8 よりグループ間で大きな差がないことから，被験者の能力差が実験結果に与えた影響は小さいと考えられる．

本実験では同一の被験者がレビューを 2 回実施していることから，1 回目のレビューで使用したレビュー技法の学習効果が，2 回目のレビューの結果に予期せぬ影響を及ぼした可能性がある．しかしながら，どちらのレビュー技法を使用する場合でもレビューの目的は同一の内容を被験者に伝えたことと，実験の手順の説明では PBR を使用する場合は表 5-11 に示したレビュー観点を用いてレビューすること，提案手法を使用する場合は 3 章で解説した手順でレビューすることを伝えており，1 回目に使用したレビュー技法の方法やレビュー観点を 2 回目のレビューの際にレビューアが考慮することは，ある程度防ぐことができたと考えられる．1 回目のレビューで PBR を使用した場合と 2 回目のレビューで PBR を使用した場合，また 1 回目のレビューで提案手法を使用した場合と 2 回目のレビューで提案手法を使用した場合のそれぞれについて，レビュー時の

考慮範囲に関するアンケートの得点の平均値差の検定を並べ替え検定により行ったところ，PBR を使用した際の「オブジェクトの悪条件」の考慮度合いは $p = 0.050$ であり 5%水準で有意差があったが，それ以外の考慮度合いには有意差はなかった．PBR を使用した際の「オブジェクトの悪条件」の考慮度合いの得点を表 14 で確認すると，2 回目のレビューよりも 1 回目のレビューの時のほうが得点は高かった．以上の結果から，「オブジェクトの悪条件」の考慮度合いに有意差があったのは 1 回目に使用したレビュー技法の学習効果によるものではなく，1 回目のレビューのレビュー対象であるパン自動会計システムの要求仕様書にオブジェクトに該当する境界条件が明示的に記述されていたことが考慮度合いに影響したものと考えられる．

4. 結言

本章では 2 つの実験により提案手法の有効性を検証した．被験者により実際に仕様書をレビューしてもらうという，製品開発の実務に近い形式で実験を実施した．実験 1 では，レビューの未経験者である学生 6 名を被験者とし，提案手法を使用して実際に，製品の改善に繋がる重要度の高い指摘を導出できることを確認した．提案手法の狙いである，レビューアの考慮範囲が広がることも確認でき，フレームワークやガイドワードが狙いどおり機能していることが確認できた．実験 2 では，実務に携わるテスト担当者 14 名を被験者とし，被験者を 7 名ずつの 2 群に分けクロスオーバー試験の形式で，PBR によるレビューと提案手法によるレビューを 2 回実施した．1 回目のレビューと 2 回目のレビューともに PBR を使用したグループよりも提案手法を使用したグループのほうが，製品の改善に繋がる重要度の高い指摘を多く導出できることを確認した．レビューの費用対効果を算出し，提案手法のほうが PBR よりも費用対効果が高いことを確認した．仕様書の記述が詳細だった 1 回目のレビューのみならず，仕様書の記述があいまいで考慮すべき境界条件が不明瞭な 2 回目のレビューにおいても提案手法は効果を得られることが確認でき，本研究の目的に示した提案手法の要件である，「(1)要求仕様書が自然言語であいまいさを含み，境界条件が不明瞭な場合でもレビューの効果を発揮できる」ことが，確認できたと言える．被験者へのアンケートおよびインタビューの回答結果から，境界条件を整理することの有効性を多くの被験者が理解し，実務でレビュー技法を活用したいと考える被験者の数は PBR と提案手法で差がないことが確認できた．

本章で確認できたのは，仮想の要求仕様書を用いた実験における提案手法の有効

性であり，実際の製品開発プロジェクトへ提案手法を適用した場合の効果は今後確認する必要がある．実際の製品開発プロジェクトへ提案手法を適用することで，レビュー技法をプロジェクトへ適用するノウハウを収集するとともに，提案手法自体のブラッシュアップや，他のレビュー技法も含めた様々なレビュー技法の活用方法の整備が今後必要である．

第６章　結論と今後の展望

1. 結論

本研究では，境界条件の整理のフレームワーク，悪条件導出のためのガイドワード，悪条件への対策案のサンプル，レビューのプロセスを定義し，悪条件分析に基づくレビュー技法を提案した．まず要求仕様のレビューにおいて考慮すべき境界条件の構造を明らかにし，境界条件を整理するフレームワークとして Boundary Breakdown Structure (BBS)を提案した．考慮すべき境界条件の要素として，「製品ライフサイクル」・「エージェント」・「ハードウェア」・「ソフトウェア（機能）」・「オブジェクト」・「外部環境」を概念モデルとして定義し，BBS のフレームワークを用いてこれらの境界条件をブレークダウンして分析できることを示した．さらにテスト分析における重要な視点である「悪条件」に着目し，その概念モデルを定義するとともに，ケーススタディに基づき悪条件導出のガイドワードを構築した．悪条件の発生源になりうる「エージェント」・「ハードウェア」・「オブジェクト」・「外部環境」の各視点からガイドワードを定義することで，要求仕様レビューへ適用しやすいガイドワードを構築した．提案手法の有効性を検証するために，要求仕様書のレビュー実験を実施した．

実験は 2 回に分けて実施し，1 回目の実験では提案手法のアプローチが正しく機能すること，すなわち提案手法のプロセスとフレームワークを使って製品の改善に繋がる重要度の高い問題をレビューアが指摘できることを確認した．具体的には，レビュー経験のない学生 6 名を被験者とし，BBS を利用した境界条件の整理，ガイドワードに基づく悪条件の導出，悪条件への対策案としての要求仕様書への指摘事項の導出を実施可能なことが確認できた．アドホックレビューに比べて，境界条件および悪条件に関する被験者の考慮範囲が広がり，重要度の高い指摘を多く導出できることを確認した．

2 回目の実験では従来のレビュー技法と比較した際の提案手法の有効性を検証した．実務に携わるテスト担当者 14 名を被験者とし，従来のレビュー技法で最も効果的とされる PBR と提案手法のそれぞれを利用したレビューの実験を行った．PBR と比較して提案手法のほうが，製品の改善に繋がる重要度の高い指摘を多く導出できることを確認した．特徴の異なる 2 つの要求仕様書をレビュー対象としたことで，提案手法を利用したレビューでは，仕様書の記述の詳細度に関わらず境界条件・悪条件を広く検討でき，重要度の高い指摘の導出が可能なことを確認した．新製品の開発時や経験の浅い

担当者が要求定義を行う場合は，要求仕様書の設計者が境界条件を明確に考慮しきれずに仕様書の記載が曖昧になりやすい．そのような場合であっても，レビューアは提案手法を利用することで設計者の想定外の悪条件を考慮して指摘を導出できる．これにより，後工程で欠陥が検出され大きな手戻りコストがかかるリスクや，製品の市場リリース後に想定外の悪条件に起因する欠陥が発生し改修コストがかかるリスクを低減することができる．提案手法は BBS や悪条件一覧といった中間成果物の作成が必要で，アドホックレビューや従来のレビュー技法よりもレビューの実施に時間を要するが，重要度の高い指摘を多く導出できた分，後工程で欠陥が検出されて修正する場合のコストを考慮するとレビューの費用対効果は PBR よりも高いことを確認した．アンケートおよびインタビューの回答結果から，実務に携わるテスト担当者に提案手法のアプローチの有効性が理解され，PBR と同等に提案手法を実務で活用したいとの回答を得た．提案手法の実務での利用可能性は PBR に劣っていないことが確認できた．

本研究の目的に提示した，(1)要求仕様書が自然言語であいまいさを含み境界条件が不明瞭な場合でもレビューの効果を発揮できること，(2)境界条件の整理と想定外の悪条件の検討のプロセスを陽に含むこと，(3)要求仕様レビューに適した境界条件の分析のフレームワークを持つこと，(4)境界条件の想定範囲を広げるアプローチをとること，という 4 つの要件を満たす個人レビュー技法として「悪条件分析に基づく要求仕様レビュー技法」を構築でき，その有効性を検証できたと言える．

2. 今後の展望

提案手法の普及に向けて，効果を得やすいレビュー対象の明確化や効果的な使い方の解説を含め，提案手法の教育方法を整備することで，レビューアが提案手法の利点を理解して使いこなせるようにする必要がある．悪条件分析を支援するソフトウェアツールを開発し，提案手法を効率的に使用できる環境を整備することも必要である．

新製品・新機能の開発時や経験の浅い担当者が要求定義を行う場合など，事前に境界条件を十分に想定することが難しい場合，提案手法により境界条件の想定範囲を広げることで効果的なレビューを実施できる．逆に，レビュー対象の境界条件が明確な場合は，PBR など従来のレビュー技法のアプローチのほうが詳細な仕様の問題点を検出しやすい．製品開発の工程やプロジェクトの状況を加味して，提案手法を適切に利用できるようにする必要がある．

レビュー対象の仕様の分量が多い場合，レビューの実施に多大な時間がかかったり，BBS が巨大になりすぎて提案手法の適用が困難になる可能性がある．この課題への対策としては，レビュー対象を機能など何らかの軸で分割し，重要な部分に絞ってレビュー技法を適用する方法が考えられる．ソフトウェアテストにおいては，検証の範囲や順序を決める手法として，リスクベースドテストが知られる[99]．不具合の発生頻度と発生時の影響の大きさからリスクを算出し，リスクの高い部分からテストを実施する．QI 手法では，仕様書文章の曖昧さに繋がる読点の多さや，文章のコピー率といった数値指標を元に，欠陥が多い可能性のある仕様書をピックアップしてレビューする[69]．製品開発プロジェクトにおける限られた時間の中で効率的に提案手法を活用するには，これらの方法の提案手法への応用や，レビューの目的に応じた他のレビュー技法と提案手法の選択や併用の方法を研究する必要がある．

提案手法はレビューの実施に従来のレビュー技法よりも多くの時間を要するため，提案手法の各工程での作業をサポートするソフトウェアツールの開発によって，提案手法によるレビューの効率を上げる必要がある．境界条件の整理・悪条件の分析を行う際に，レビューアの思考に合わせて適切なフレームワークやガイドワードを提示し分析をサポートするソフトウェアツールを開発すれば，効率的に提案手法を利用できる．過去のレビューで導出された境界条件や悪条件，レビューで見逃されて問題となった境界条件や悪条件のデータを蓄積し，提案手法を利用する際に，レビュー対象に関連する境界条件や悪条件をソフトウェアツールによりサジェストできれば，境界条件や悪条件の考慮漏れを減らし，効果的なレビューを行うことが可能となる．

境界条件を整理するアプローチは，レビューだけでなく要求仕様書の作成時に境界条件の想定範囲を可視化・構造化するためにも利用できると考えられる．要求仕様書を作成する段階で設計者自身が BBS を用いて境界条件の整理・悪条件の検討を行うことで，境界条件の考慮漏れを減らすことが可能となる．要求仕様書を作成する段階での提案手法の適用方法や，適用した場合の効果の検証は今後の課題である．

提案手法はレビューアによって導出できた指摘の件数のばらつきが大きく，レビューアによっては効果を得にくいと考えられる．提案手法をうまく利用するには境界条件を構造化する能力や悪条件を想定する能力が必要である．これらの能力は要求分析やテスト分析などの業務経験を経て培われると考えられるが，やみくもに業務を実施しても能力は伸びにくい．提案手法は境界条件を整理するプロセスを明示的に含むため，

提案手法を利用してレビューすることやレビュー結果についてプロジェクトのメンバー間で議論することは，レビューアにとって，境界条件を構造化する能力や悪条件を想定する能力の訓練になると考えられる．提案手法を利用することによる教育効果は，実際の製品開発プロジェクトで今後確認していく必要がある．

本研究ではレビューで導出した指摘事項の取り扱い方法について議論していない．実際の製品開発プロジェクトでは，レビューで導出された指摘事項 1 件ずつに対して，仕様書の修正が必要かどうか議論・検討し，必要な修正を仕様書に加え，最終的な仕様書の内容に基づきステークホルダーと合意形成する必要がある．提案手法により導出された指摘を製品の改善に最終的に活かすには，指摘の取り扱いの方法やプロセスを設計する必要がある．ゲーミングの研究分野では，ゲームを実施した後に司会・参加者によりディブリーフィングを行うことの重要性が知られている．ディブリーフィングではゲームを通じて体験したことを振り返り，ゲームから得られた学びについて議論が行われる．ディブリーフィングの役割として，参加者同士の議論によって現実認識が複数存在することに気付く役割や，自分にとっての現実認識と他者にとっての現実認識が異なるものであることを理解するといった役割が知られる[100]．仕様のレビュー後に設計者とレビューアで仕様の修正要否を議論する際や，ステークホルダー間で仕様の内容への合意形成を得るための議論を行う際に，ゲーミングにおけるディブリーフィングの方法を応用できる可能性があり，今後さらなる研究が必要である．

今後，実際の製品開発プロジェクトでの提案手法の活用を進めながら，従来のレビュー技法の活用も含めた効果的なレビューの方法を研究していく．さらに，提案手法を核とした，仕様の設計とステークホルダー間での合意形成の手法を体系化するとともに，組織におけるエンジニアの教育に提案手法を活用する仕組みを構築し，製品開発の上流工程における品質の作りこみに貢献していく．

参考文献

[1] Checkland, P. 著, 高原康彦・中野文平 監訳, 新しいシステムアプローチ―システム思考とシステム実践―, 株式会社オーム社, 1985.

[2] March, S. T., and Smith, G. F., “Design and natural science research on information technology,” Decision Support Systems, 15(4), 251-266, 1995.

[3] Hevner, A. R., March, S. T., Park, J., and Ram, S., “Design Science in Information Systems Research,” MIS Quarterly, 28(1), 75-105, 2004.

[4] Islam, M. S., and Grönlund, Å., “Applying Design Science Approach in ICT4D Research,” Practical Aspects of Design Science. EDSS 2011. Communications in Computer and Information Science, 286, 132-143, 2012.

[5] Baskerville, R., Kaul, M., Pries-Heje, J., and Storey, V., “Inducing Creativity in Design Science Research,” Extending the Boundaries of Design Science Theory and Practice. DESRIST 2019. Lecture Notes in Computer Science, 11491, 3-17, 2019.

[6] 佐々木眞一, 自工程完結-品質は工程で造りこむ, 日本規格協会, 2014.

[7] 大島正善,「ソフトウェア・プロジェクトの上流工程における品質保証活動と定量的価値」, プロジェクトマネジメント学会 2000 年度秋季研究発表大会予稿集, 62-65, 2000.

[8] 情報サービス産業協会 REBOK 企画 WG 編, 要求工学知識体系 第 1 版, 近代科学社, 2011.

[9] 独立行政法人情報処理推進機構　社会基盤センター監修, ソフトウェア開発分析データ集 2020, 2020.

[10] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J. and Thomas, D., Manifesto for Agile Software Development, https://moodle2019-20.ua.es/moodle/pluginfile.php/2213/mod_resource/content/2/agile-manifesto.pdf（参照 2021-1-21）.

[11] 谷﨑浩一, 田上諭, 森龍二, 蛭田恭章, 森崎修司,「アジャイル開発に適した品質チェック項目の作成手法」, ソフトウェアエンジニアリングシンポジウム 2020 論文集, 88-96, 2020.

[12] 谷﨑浩一, 田上諭, 森龍二, 蛭田恭章, 森崎修司,「アジャイル開発に適した品質チェック項目の構造化」, ソフトウェア品質シンポジウム 2020 報文集, B1-1, 2020.

[13] 森崎修司, 久保匡, 荻野利彦, 阪本太志, 山田淳,「過去の不具合の修正工数を考慮したソフトウェアレビュー手法」, 電子情報通信学会論文誌 D, 95(8), 1623-1632, 2012.

[14] リンダ・M・ライルド (著), M・キャロル・ブレナン (著), 野中誠 (翻訳), 鷲崎弘宜 (翻訳), 演習で学ぶソフトウエアメトリクスの基礎 ソフトウエアの測定と見積もりの正しい作法, 日経 BP, 2009.

[15] 独立行政法人情報処理推進機構, 「「コストモデル」を使った開発品質・生産性向上の取組み ～バグ対応コストの見える化と最適化～」, 先進的な設計・検証技術の適用事例報告書 2015 年版 PARTⅢ 検証事例, 2015.

[16] 独立行政法人情報処理推進機構, 2018 年度組込みソフトウェア産業の動向把握等に関する調査事業 「組込み/IoT に関する動向調査」調査報告書, 2019.

[17] 独立行政法人情報処理推進機構, 2018 年度組込み/IoT に関する動向調査 調査報告書（データ編）, 2018.

[18] 小川秀人, 市井誠, 新原敦介, 鈴木康文, Phan Thi Thanh Huyen, 坂井田真也,「形式手法のソフトウェア開発利用の課題と事例」, ウィンターワークショップ 2015・イン・宜野湾 論文集, 43-44, 2015.

[19] 山田隆弘, 「形式手法を開発現場に持ち込むには -試験計画より始めよ-」, ウィンターワークショップ 2016・イン・逗子 論文集, 7-8, 2016.

[20] 青木利晃, 「車載システム開発における形式手法実践の現状と課題」, システム／制御／情報, 62(4), 134-140, 2018.

[21] 独立行政法人情報処理推進機構 社会基盤センター監修, ソフトウェア開発データ白書 2018-2019, 2018.

[22] ISO/IEC 20246:2017. Software and systems engineering — Work product reviews.

[23] Adrion, W. R., Branstad, M. A., and Cherniavsky, J. C., “Validation, Verification, and Testing of Computer Software,” ACM Computing Surveys, 14(2), 155-192, 1982.

[24] Shull, F., “Developing Techniques for Using Software Documents: A Series of Empirical Studies,” PhD thesis, Computer Science Dept., Univ. of Maryland, 1998.

[25] Shull, F., Rus, I., and Basili, V., “How perspective-based reading can improve requirements inspections,” Computer, 33(7), 73-79, 2000.

[26] 出口弘, 「社会シミュレーション&サービスシステムが目指す世界」, 計測と制御, 52(7), 563-567, 2013.

[27] 西康晴, テスト観点に基づくテスト開発方法論 VSTeP の概要, http://qualab.jp/materials/VSTeP.130510.color.pdf（参照 2021-1-21）.

[28] 秋山浩一, 事例とツールで学ぶ HAYST 法―ソフトウェアテストの考え方と上達のポイント, 日科技連出版社, 2014.

[29] Yumoto, T., Matsuodani, T., and Tsuda, K., “A Test Analysis Method for Black Box Testing Using AUT and Fault Knowledge,” Procedia computer science, 22, 551-560, 2013.

[30] Tanizaki, K., Hiruta, Y., Soma, T., Yamao, N., Kato, S., and Iizuka, Y., “A Method for Software Test Design Considering Weakness and Adverse Condition,” in Proc. of The 18th ANQ Congress 2020, 536-548, 2020.

[31] Eero J. Uusitalo, Marko Komssi, Marjo Kauppinen, Alan M. Davis, “Linking Requirements and Testing in Practice,” in Proc. of 16th IEEE International Requirements Engineering Conference, 265-270, 2008.

[32] 山本修一郎, 「要求とテスト」, ビジネスコミュニケーション, 46(4), 50-54, 2009.

[33] Bjarnason, E., and Borg, M., “Aligning Requirements and Testing: Working Together toward the Same Goal,” IEEE Software, 34(1), 20-23, 2017.

[34] Nielsen, J. and Landauer, T. K., “A mathematical model of the finding of usability problems,” in Proc. of the INTERACT '93 and CHI '93 Conference on human factors in computing systems, 206-213, 1993.

[35] Martin, J., and Tsai, W., “N-Fold Inspection： a requirements analysis technique,” Communications of the ACM, 33(2), 225-232, 1990.

[36] 沖汐大志, 誉田直美, 森田純恵, 大場みち子, 小島嘉津江, 服部克己, 藤原良一, 「レビュー技術動向」, SQuBOK 策定部会編　SQuBOK Review 2017, 日本科学技術連盟, 1-18, 2017.

[37] Ciolkowski, M., Laitenberger, O. and Biffl, S., “Software reviews, the state of the practice,” IEEE Software, 20(6), 46-51, 2003.

[38] 森崎修司, なぜ重大な問題を見逃すのか？ 間違いだらけの設計レビュー改訂版, 日経 BP 社, 2015.

[39] 森崎修司, 松本健一, 「業務観点でのレビューを目指した不具合情報の分析」, 情報処理学会研究報告, 2013-SE-179(35), 1-8, 2013.

[40] Kasubuchi, K., Morisaki, S., Yoshida, A., and Ogawa, C., An empirical evaluation of the effectiveness of inspection scenarios developed from a defect repository, in Proc. of 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME), 439-448, 2015.

[41] Ureshino, A., Nishi, Y., Yoshizawa, S., Nonaka, M., and Asai, M., “Trap-Based Software Review: How Will Engineers Embed Bugs Due To Cognitive Error?,” in Proc. of 2018 IEEE International Symposium on Software Reliability Engineering Workshops, 6-11, 2018.

[42] Basili, V., Green, S., laitenberger, O., Shull, F., Sorumgard, S. and Zelkowitz, M. V., “The Empirical Investigation of Perspective-Based Reading”, Empirical software engineering : an international journal., 1(2), 133-164, 1996.

[43] 岡本博幸, 内山敬太, 鈴木彩子, 高橋一仁, 西山茂, 野中誠, 「要求仕様書の特性に着目した個人レビュー手法の実験的評価」, http://www.juse-sqip.jp/workshop/seika/2004/6/6_report.pdf（参照 2021-1-21）.

[44] Thelin, T., Runeson, P., and Wohlin, C., “An experimental comparison of usage-based and checklist-based reading,” IEEE Transactions on Software Engineering, 29(8), 687-704, 2003.

[45] Shull, F., “Developing Techniques for Using Software Documents: A Series of Empirical Studies,” PhD thesis, Computer Science Dept., Univ. of Maryland, 1998.

[46] Biffl, S., Software Inspection Techniques to Support Project and Quality Management, Shaker, 2001.

[47] Halling, M., Biffl, S., Grechenig, T. and Ko¨hle, M., “Using Reading Techniques to Focus Inspection Performance,” in Proc. of 27th Euromicro Workshop Software Process and Product Improvement, 248-257, 2001.

[48] Miller, J., Wood, M. and Roper, M., “Further Experiences with Scenarios and Checklists,” Empirical software engineering : an international journal, 3(3), 37-64, 1998.

[49] Sandahl, K., Blomkvist, O., Karlsson, J., Krysander, C., Lindvall, M. and Ohlsson, N., “An Extended Replication of an Experiment for Assessing Methods for Software Requirements,” Empirical software engineering : an international journal, 3(4), 381-406, 1998.

[50] Porter, A., Votta, L. and Basili, V. R., “Comparing Detection Methods for Software Requirements Inspection: A Replicated Experiment,” IEEE transactions on software engineering, 21(6), 563-575, 1995.

[51] Porter, A. and Votta, L., “Comparing Detection Methods for Software Requirements Inspection: A Replication Using Professional Subjects,” Empirical software engineering : an international journal, 3(4), 355-380, 1998.

[52] Wakabayashi, T., Morisaki, S., Kasai, N., Atsumi, N., and Yamamoto, S., “Tool Supported Detection of Omissions by Comparing Words between Requirements and Design Document,” Journal of Information Processing, 28, 136-149, 2020.

[53] Bacchelli, A., and Bird, C., “Expectations, Outcomes, and Challenges of Modern Code Review,” in Proc. of the 2013 International Conference on Software Engineering, 712-721, 2013.

[54] GitHub, https://github.com（参照 2021-1-21）.

[55] ISTQB, ISTQB Worldwide Software Testing Practices Report 2017-18, https://www.istqb.org/documents/ISTQB%202017-18_Revised.pdf（参照 2021-1-21）.

[56] 大村朔平，一般システムの現象学―よりよく生きるために，技報堂出版，2005.

[57] Yumoto, T., Matsuodani, T., and Tsuda, K., “A study on an approach for analysing test basis using I/O test data patterns,” in Proc. of 2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops, 1-8, 2015.

[58] 湯本剛，ゆもつよメソッドにおけるテスト分析とテスト設計，https://www.slideshare.net/yumotsuyo/ss-71502350（参照 2021-1-21）.

[59] 安達賢二, 「レビュー目的・観点設定の効果と課題」，ソフトウェアテストシンポジウム 2016 東京 予稿集, A2-1, 2016.

[60] 西康晴, QA アーキテクチャの設計による説明責任の高いテスト・品質保証, https://www.slideshare.net/YasuharuNishi/qa-67559281（参照 2021-1-21）.

[61] Despotou, G. and Kelly, T., “Extending the Safety Cases Concept to Address Dependability,” in Proc. of the 22ndInternational System Safety Conference, 645-654, 2004.

[62] Kelly, T., “Arguing Safety - A Systematic Approach to Managing Safety Cases,” DPhil Thesis, Department of Computer Science, University of York, York, 1998.

[63] 森崎修司, ソフトウェアレビュー入門（3）：“読み方”を知って、レビューをもっと効果的に，https://www.itmedia.co.jp/im/articles/0911/19/news106_4.html（参照 2021-1-21）.

[64] リー・コープランド（著），宗雅彦（訳），はじめて学ぶソフトウェアのテスト技法, 日経 BP, 2013.

[65] International Software Testing Qualifications Board, ISTQB テスト技術者資格制度 Foundation Level シラバス 日本語版 Version 2018V3.1.J02, http://jstqb.jp/dl/JSTQB-SyllabusFoundation_Version2018V31.J02.pdf （参照 2021-1-21）.

[66] 近江久美子, 永田敦, 阿部修久, 天野佑太, 伊藤慈朗, 上田克則, 朝永糸子, 森崎一邦,「要求仕様書におけるテストエンジニアの視点を活かした欠陥検出方法の提案」，ソフトウェア品質管理研究会 第 24 年度（2008 年度）分科会成果報告, 2009.

[67] 細川宣啓, The Art of Quality Inspection - 品質インスペクション技術の全容とその効果 -, https://www.ipa.go.jp/files/000004723.pdf（参照 2021-1-21）.

[68] 高橋 功, 上田 裕之, 高橋 実雄, 中谷 一樹, 細川 宣啓, 永田 敦, 藤原 雅明, 森崎 修司, 「HDR 法 : 仮説駆動型レビュー手法の提案－HDR 法の実践による生産性と品質の同時向上」, ソフトウェア品質シンポジウム 2013 報文集, A1-2, 2013.

[69] 細川宣啓, 「第三者インスペクションによる品質検査と欠陥予防」, 情報処理, 50(5), 405-411, 2009.

[70] IEC 61882:2016. Hazard and operability studies (HAZOP studies) — Application guide.

[71] Leveson, N. G., Engineering a Safer World: Systems Thinking Applied to Safety, MIT Press, 2012.

[72] Pumfrey, D. J., “The Principled Design of Computer System Safety Analyses,” PhD thesis, The University of York, 1999.

[73] 河野哲也, 「ソフトウェア要求仕様における HAZOP を応用したリスク項目設計法」, ソフトウェアテストシンポジウム 2012 東京 予稿集, D2-1, 2012.

[74] 安達賢二, 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！, http://www.jasst.jp/symposium/jasst16shikoku/pdf/S2.pdf（参照 2021-1-21）.

[75] 大林英晶, 森崎修司, 渥美紀寿, 山本修一郎, 「逸脱分析を用いた要求仕様書からのテスト項目抽出手法」, 情報処理学会論文誌, 57(4), 1262-1273, 2016.

[76] 出口弘, 人工物の科学方法論―その範囲と認識論的基盤―, http://phsc.jp/dat/rsm/20141023_b-1.pdf（参照 2021-1-21）.

[77] 出口弘, 「社会システムの研究動向 2−研究のための方法論− 社会システムの制度デザインの方法論：政策科学の方法としてのエージェントベースモデリング&シミュレーション」, 計測と制御, 52(7), 574-581, 2013.

[78] 三瀬敏朗, 新屋敷泰史, 橋本正明, 鵜林尚靖, 片峯恵一, 中谷多哉子, 「組込みソフト非正常系における基礎モデル及び仕様分析手法の提案」, 情報処理学会研究報告, 2004-SE-147(29), 81-88, 2005.

[79] アリスター・コーバーン（著）, ウルシステムズ株式会社（監訳）, ユースケース実践ガイド―効果的なユースケースの書き方, 翔泳社, 2001.

[80] 独立行政法人情報処理推進機構, 情報セキュリティマネジメントと PDCA サイクル, https://www.ipa.go.jp/security/manager/protect/pdca/risk.html（参照 2021-1-21）.

[81] Alting, D. L. and Jogensen, D. J., “The Life Cycle Concept as a Basis for Sustainable Industrial Production,” CIRP Annals, 42(1), 163-167, 1993.

[82] Breiing, A., Engelmann, F. and Gutowski, T., “Engineering Design,” in Springer Handbook of Mechanical Engineering, K.–H. Grote and E. K. Antonsson, Ed. Springer, 819-877, 2009.

[83] 樽本徹也, ユーザビリティエンジニアリング　第 2 版, オーム社, 2014.

[84] Leveson, N. G. and Thomas, J. P., STPA HANDBOOK, https://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf（参照 2021-1-21）.

[85] 水本篤,「サンプルサイズが小さい場合の統計的検定の比較―コーパス言語学・外国語教育学への適用―」, 統計数理研究所共同研究リポート 238 「言語コーパス分析における数理データの統計的処理手法の検討」, 1-14, 2010.

[86] Biffl, S., and Halling, M., “Investigating the influence of inspector capability factors with four inspection techniques on inspection performance,” in Proc. of Eighth IEEE Symposium on Software Metrics, 107-117, 2002.

[87] 折笠秀樹, 「クロスオーバー試験の計画および解析」, 薬理と治療, 44(9), 1261-1276, 2016.

[88] Simpson, T. W., “Experimental Methods and Human Nutrition,” Supplement to the Journal of the Royal Statistical Society, 5(1), 46-69, 1938.

[89] Cochran, W. G., Antrey, K. M. and Cannon, C. Y., “A Double Change-Over Design for Dairy Cattle Feeding Experiments,” Journal of Dairy Science, 24(11), 937-951, 1941.

[90] Canadian Task Force on the Periodic Health Examination, “The periodic health examination,” Canadian Medical Association journal, 121(9), 1193-1254, 1979.

[91] U.S. Preventive ServicesTask Force, “Guide to Clinical Preventive Services. 2nd edition,” Williams & Wilkins, 1996.

[92] 折笠秀樹,「臨床研究のエビデンスレベル」, 内分泌・糖尿病・代謝内科, 45(2), 104-109, 2017.

[93] Oxford Centre for Evidence-Based Medicine, Levels of Evidence (March 2009), https://www.cebm.ox.ac.uk/resources/levels-of-evidence/oxford-centre-for-evidence-based-medicine-levels-of-evidence-march-2009（参照 2021-2-22）.

[94] 佐藤俊哉, 「人間栄養学講座 連載 ランダム化臨床試験をする前に 【第 3 回】ランダム化臨床試験のデザイン」, 栄養学雑誌, 65(4), 193-197, 2007.

[95] 森崎修司, 技法の分類とテンプレート（応用編）, https://thinkit.co.jp/article/893/1?page=0%2C1（参照 2021-1-21）.

[96] 小池利和, 相澤武, 中川忠彦, 渡井克己, 山下英祐, 奥村学, 「レビューの質と価値の定量化の提案」, ソフトウェア品質管理研究会　第 23 年度（2007 年度）分科会成果報告, 2008.

[97] 組込みソフトウェア管理者・技術者育成研究会, 話題沸騰ポット(GOMA-1015 型) 要求仕様書 第 7 版, http://www.sessame.jp/workinggroup/WorkingGroup2/POT_Specification_v7.PDF（参照 2021-1-21）.

[98] ソフトウェアテスト技術振興協会, テスト設計コンテスト'15 テストベース「自動販売機プログラム（ASTER オリジナル）」, http://aster.or.jp/business/contest/doc/2015tdc-v1_1.zip（参照 2021-1-21）.

[99] Veenendaal, E., Practical risk-based testing: the PRISMA approach, UTN Publishers, 2012.

[100] 中村美枝子, 「多元的現実の理解を促すディブリーフィング ―仮想世界ゲームに基づく提案―」, シミュレーション＆ゲーミング, 18(1), 21-30, 2008.

謝辞

本研究を行うにあたり，日ごろからご指導くださった出口弘教授に深く感謝いたします．東京工業大学と縁もゆかりもなかった私が博士課程に通わせていただいて，この論文を仕上げられたのは，ひとえに出口先生のおかげです．ジャーナルへ投稿した論文がなかなか通らずに苦しんでいた時期も，根気強く修正のための議論をさせていただき，本当にありがとうございました．

助教のチャンシュアン先生には研究を進めるうえでの注意点や中間発表・予備審査などの折々の学事がどのようなものか教えていただき，ありがとうございました．博士課程の研究という自分にとって未知の取り組みを進めるうえでの道標になりました．

研究室秘書の林原雅美さんには，さまざまな事務手続きにおいてお世話になり，ありがとうございました．物凄い速さでメールのご返信をいただけて，いつも助かりました．事務手続きで困ったら，林原さんに伺えば何とかなるという安心感がありました．

出口研究室は社会人博士が多く，ゼミ・合宿・学会発表などの際に情報交換をさせていただき，とても有意義でした．また，研究室に物理的に居る時間の短い社会人博士にとって，修士の学生の皆さまとはお話する機会が少なかったですが，たまに研究室に行った際や，合宿などのイベントの際に気さくに話しかけてもらって，嬉しかったです．

株式会社ベリサーブの蛭田恭章さん，マレリ株式会社の相馬武さんには，研究に行き詰っているときに相談や雑談をさせていただき，ありがとうございました．本論文の第 5 章に記載した実験 2 は蛭田さんの助力なしでは実現できませんでした．この場を借りて御礼申し上げます．実験に参加してくださった皆さまに対しても感謝の念に堪えません．

株式会社ベリサーブの研究企画開発部ならびに関係部署の皆さまに感謝いたします．たびたび不在でご迷惑をかけることもあったかと思いますが，柔軟に勤務を調整させてもらえたことで，仕事を続けながら博士課程に通うことができました．特に上長の「今は博士課程を一番優先すべき．最悪，仕事は休んでいいからね」という言葉には精神的にとても救われ，安心して研究や論文執筆に取り組むことができました．今後は会社の研究開発活動をさらに加速させていきたいと思います．

その他，ここには書ききれないほど多くの方々に支えられて，博士課程の研究を進めてこの論文を完成させることができました．この場を借りて謹んで御礼申し上げます．

そして最後に，両親と弟，祖父母，いつも近くで支えてくれた妻に感謝いたします．

業績リスト

論文誌（査読あり）

[1] 谷崎浩一, Shuang, C.・出口弘,「要求仕様レビュー技法のクロスオーバー試験に基づく評価研究」, シミュレーション＆ゲーミング, 30(2), 95-109, Dec. 2020.

国際会議会議録掲載論文（査読あり）

[2] Koichi Tanizaki, Shuang Chang and Hiroshi Deguchi. “Propose a Method of Requirements Review considering Adverse Conditions,” in Proc. of 2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS), 383-388, Dec. 2018.

国際会議発表（査読あり）

[3] Koichi Tanizaki, Shuang Chang, Hiroshi Deguchi. “Propose a Method to Provide Feedback for Improving Requirements from the Viewpoint of Test Analysis”, The 10th AESCS International Workshop, Mar. 2018.

国内会議会議録掲載論文（査読あり）

[4] 谷崎浩一・Shuang, C.・出口弘,「要求仕様書の改善のためのフィードバック導出手法」, 日本シミュレーション＆ゲーミング学会　全国大会論文報告集 2018 年秋号, 30-37, Nov. 2018.

口頭発表

[5] 谷崎浩一, 出口弘.「テスト分析の視点から要求の改善を促す方法論の構築」, 計測自動制御学会　第 14 回社会システム部会研究会, Jul. 2017.

Appendix A　パン自動会計システム要求仕様書

実験で使用したパン自動会計システムの要求仕様書を掲載する．実際の製品を参考に筆者が作成した，本研究オリジナルの仕様書である．

パン自動会計システム（仮称）
要求仕様書

Ver.1.0

2018年5月2日

TEVASAKIベーカリー

更新履歴

バージョン	作成・更新日	改版理由・箇所	担当
1.0	2018/5/2	初版作成	谷崎

1. 本文書の位置づけ

本文書はパン自動会計システム（仮称）の要求仕様書として、システムの概要・必要な機能要件・画面イメージを記したものである。
パン自動会計システム（仮称）はTEVASAKIベーカリーの各店舗にて、会計業務を行う際に、レジ係の店員によって使用されるものである。

2. システム導入の目的

パン自動会計システム（仮称）は、TEVASAKIベーカリーの各店舗のレジにおける会計業務を、効率化することを目的とする。
店員がパンの値段を覚えたり調べる必要がなくなり、レジ打ちの時間の短縮に繋がる。

3. システムの概要

3-1.全体イメージ



図1.パン自動会計システム（仮称）全体イメージ

3-2.ハードウェア

・カメラユニット
・レジ機＋店員側ディスプレイ＋購入者側ディスプレイ
※店員側ディスプレイはタッチパネル
・トレイ
※専用のトレイを用意しなくても、既定の範囲の白色のトレイであれば利用可能とする

3-3.OS

・レジ機のOSはWindows 10 Pro 64bitとする

3-4.インタフェース

・レジ機とカメラユニットのインタフェースはUSB 3.0とする
※カメラユニットの電源も兼用
・レジ機と店員側ディスプレイ・購入者側ディスプレイのインタフェースはHDMIとする

4. 機能要件

4-1.店員用機能

(1)パンの商品名と単価を登録する
・パン種別・単価登録画面でパンの種別と単価を登録する
・パンを10回撮影し、パンの商品名と単価をキーボードで入力し、登録する
・登録可能な商品名は100文字以内、単価は0円～999円の範囲とする。
・登録済みの商品名を登録しようとすると、エラーが表示され登録できない
・100種類まで登録できる

(2)登録済みのパンの商品名と単価を表示・検索する

・パンの商品名、単価を一覧表示できる
・パンの商品名でフィルタリングできる。フィルタリングは文字列の部分一致とする。
文字を入力するたびにフィルタリングされる。
・パンの商品名と単価でソートできる

(3)登録済みのパンの商品名と単価を更新・削除する
・パンの商品名、単価を編集できる
・登録した情報を削除できる

(4)カメラでパンを撮影しパンの種類を識別する
・撮影ボタンを押すと撮影される
・撮り直しできる
・パンとして認識された物体は緑色に縁取りされ、縁取りの中央に商品名と単価が
表示される
・パンの識別にかかる時間は、パン1個につき0.2秒以内とする
・パンの識別の正答率は90%を目標とする

(5)パンを正しく識別できない場合、ユーザがパンの種類を指定する
・認識が間違っている可能性がある場合、物体は赤色に縁取りされる
・赤色に縁取りされた物体をタッチすると、登録済みのパンの商品名と単価が
一覧表示され、その中からパンの種類を選択できる。表示される順番は、
候補として可能性が高いものから順に表示する。
・商品名と単価の一覧表は、パンの商品名でフィルタリングできる。
フィルタリングは文字列の部分一致とし、文字を入力するたびにフィルタリングされる
・商品名を選択すると縁取りが黄色になり、縁取りの中央に選択した名称が表示される

(6)パンの種類を変更したり会計対象から除外する
・縁取りされた物体をタッチすると、登録済みのパンの商品名と単価が一覧表示され、
その中からパンの種類を選択できる。
・商品名と単価の一覧表は、パンの商品名でフィルタリングできる。
フィルタリングは文字列の部分一致とし、文字を入力するたびにフィルタリングされる
・パン以外の物体がトレイに乗っている場合、会計の対象から除外できる

(7)一覧の中に候補が見つからない場合、単価を手入力する
・画面上のテンキーをタッチすることで単価を入力できる
・入力可能な単価は0円～999円の範囲とする
・単価を手入力した場合、パンの名称は「ﾊﾟﾝ」とする

(8)合計金額を計算する
・トレイ上のすべてのパンに黄色か緑色の縁取りが付くと、決定ボタンが活性化される
・決定ボタンを押すと、認識されたパンの商品名・単価・個数・金額が一覧表示される
・商品名・単価・個数・金額の一覧表は、商品名の昇順で表示される
・金額は、単価×個数の値が表示される
・小計・消費税・税込合計の計算結果が表示される。消費税は1円未満を切り捨てて、
整数で表示する。
・各値の範囲は以下とする。
個数：0個～99個
単価：0円～999円
金額：0円～99,999円
小計：0円～999,999円
消費税：0円～999,999円
税込合計：0円～999,999円
・消費税が変わる可能性を考慮し、税率を変更する手段を持つ

4-2.購入者用機能
(1)店員側ディスプレイの画面に応じて、顧客側ディスプレイの画面を更新する
※店員側ディスプレイと顧客側ディスプレイの画面の対応は、「5.画面イメージ」の
画面一覧を参照

(2)準備中画面を表示する
・レジが準備中であることを知らせるために、準備中画面を表示する

(3)待ち受け画面を表示する
・会計を行っていない際は、待ち受け画面を表示する
・待ち受け画面に表示する画像は、店員側ディスプレイのメニュー画面から変更できる

(4)パン識別結果・会計画面を表示する
・パンの識別結果と合計金額を表示する

4-3.共通

(1)機器の起動・終了
・電源ボタンを押すことで起動する
・シャットダウンボタンを押すことでシャットダウンする
・起動・シャットダウンは、OSの機能によって行う

(2)数値表示のルール
・数値を表示する際は、ゼロパディングしない
例）120円の場合「120」と表示する。「0120」とはしない
・金額を表示する際は、3ケタごとにカンマを打つ
例）1200円の場合「1,200」と表示する

(3)性能
各画面の遷移は1秒以内とする。ただし、パンの識別にかかる時間は除く。

5.画面イメージ

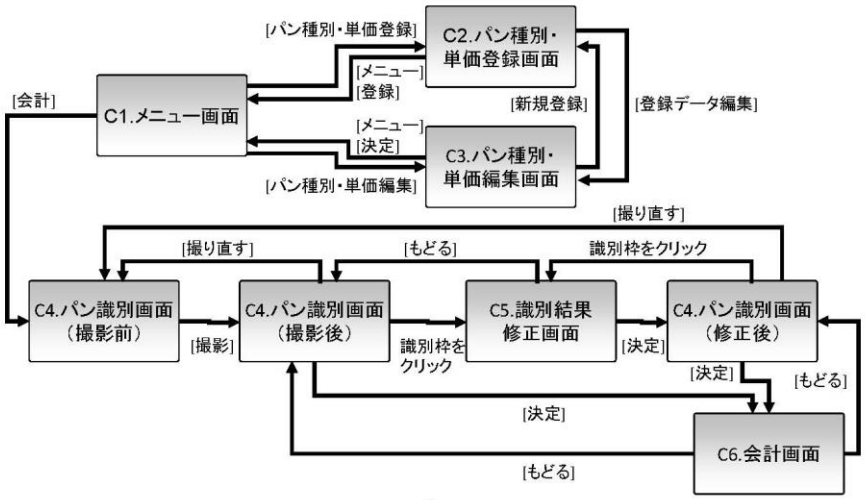
次ページ以降に画面イメージを示す

画面一覧

※顧客側ディスプレイは店員側ディスプレイの画面遷移に連動する

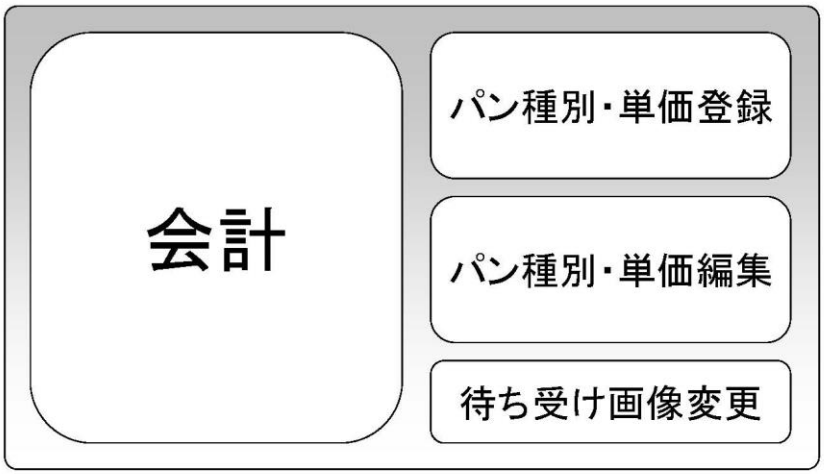
店員側ディスプレイ	顧客側ディスプレイ
C1.メニュー画面	P1.準備中画面
C2.パン種別・単価登録画面	
C3.パン種別・単価編集画面	
C4.パン識別画面（撮影前）	P2.待ち受け画面
C4.パン識別画面（撮影後）	
C5.識別結果修正画面	
C4.パン識別画面（修正後）	
C6.会計画面	P3.パン識別結果・会計画面

画面遷移（店員側ディスプレイ）

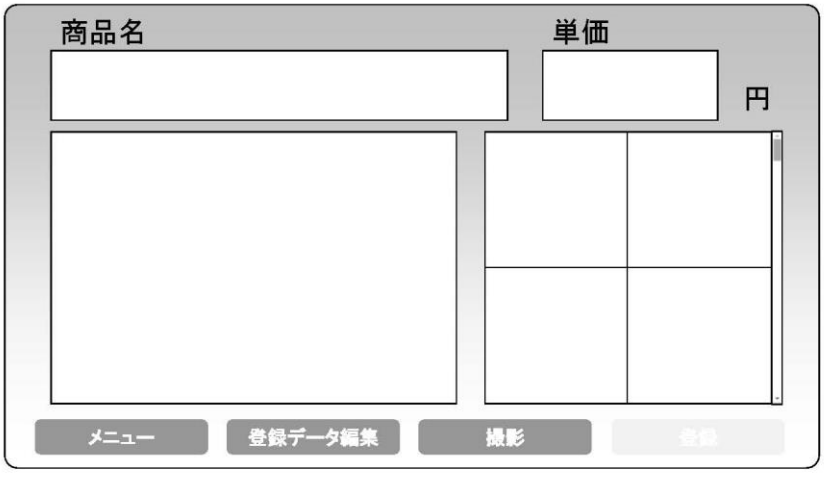


6

店員側ディスプレイ
C1.メニュー画面



店員側ディスプレイ
C2.パン種別・単価登録画面



店員側ディスプレイ
C2.パン種別・単価登録画面



店員側ディスプレイ
C3.パン種別・単価編集画面

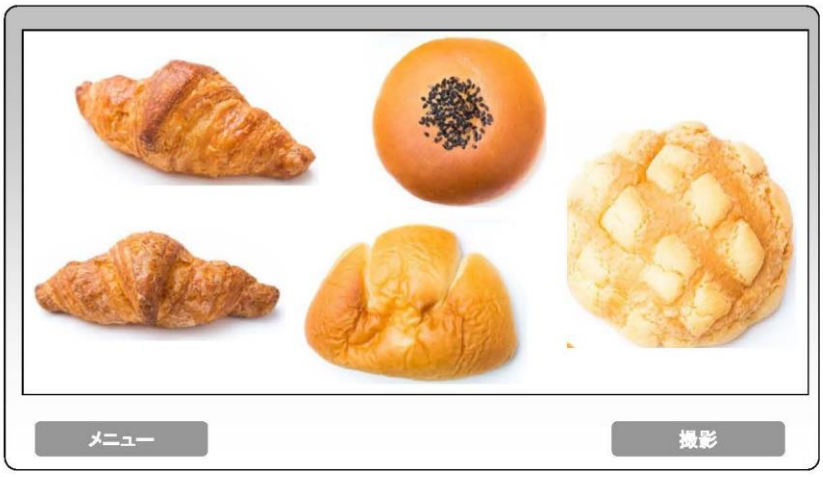


8

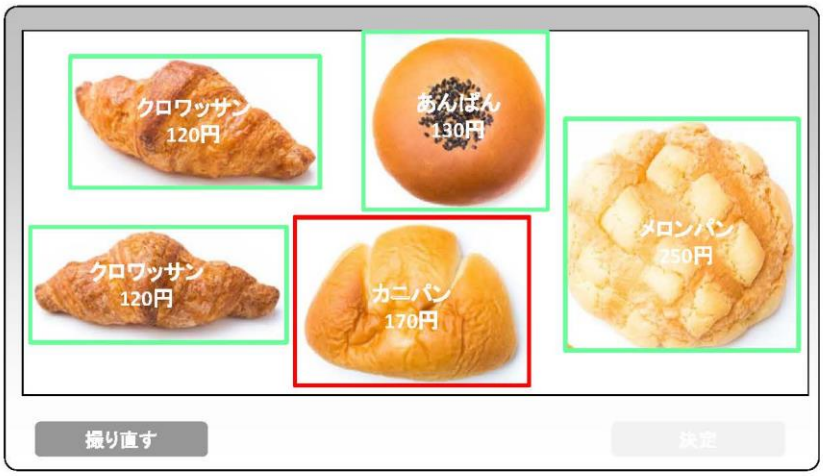
店員側ディスプレイ
C3.パン種別・単価編集画面



店員側ディスプレイ
C4.パン識別画面（撮影前）



店員側ディスプレイ
C4.パン識別画面（撮影後）



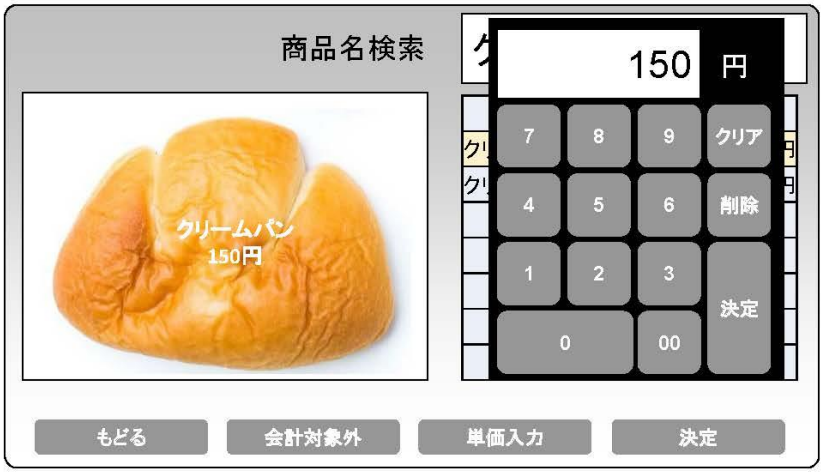
店員側ディスプレイ
C5.識別結果修正画面



店員側ディスプレイ
C5.識別結果修正画面（フィルタリング）

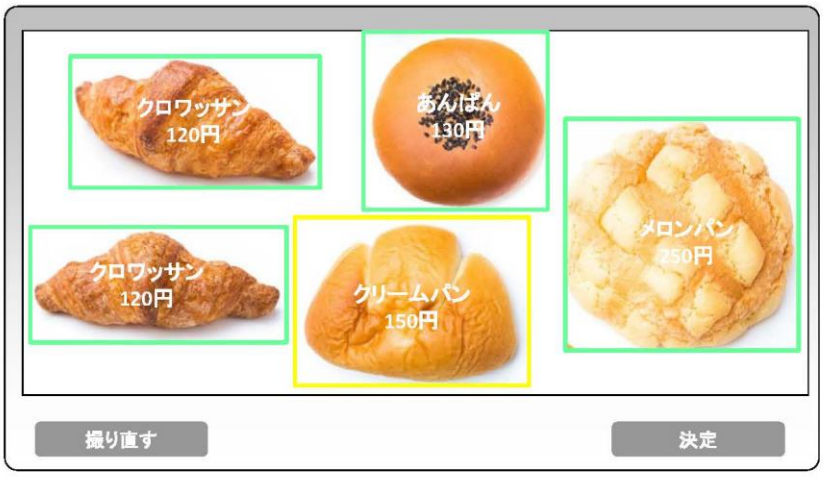


店員側ディスプレイ
C5.識別結果修正画面（単価入力）

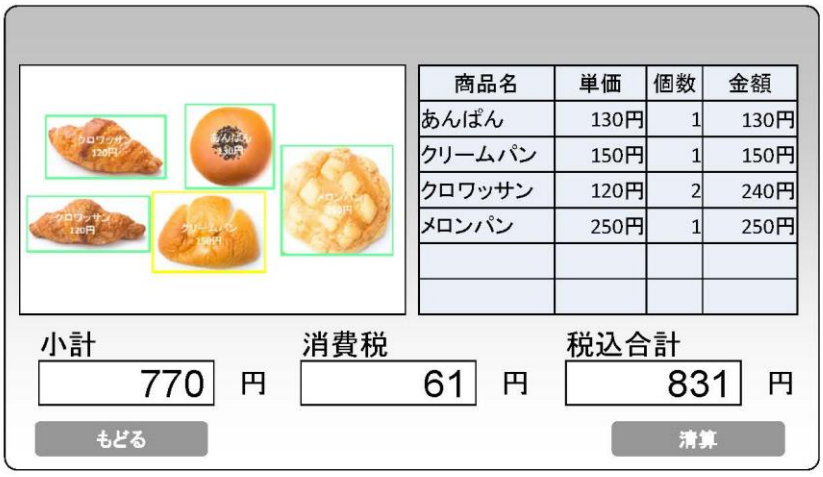


11

店員側ディスプレイ
C4.パン識別画面（修正後）



店員側ディスプレイ
C6.会計画面



12

購入者側ディスプレイ
P1.準備中画面

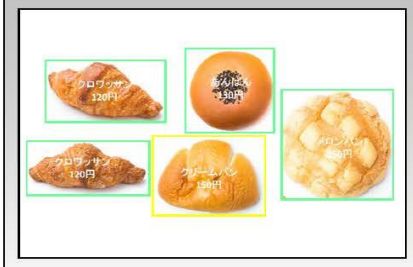
準備中
しばらくお待ちください

購入者側ディスプレイ
P2.待ち受け画面（サンプル）



13

購入者側ディスプレイ
P3.パン識別結果・会計画面



商品名	単価	個数	金額
あんぱん	130円	1	130円
クリームパン	150円	1	150円
クロワッサン	120円	2	240円
メロンパン	250円	1	250円

小計 770 円　消費税 61 円　税込合計 831 円

14

Appendix B　提案手法のフレームワークとガイドワード

実験で使用した BBS のフレームワークを図 B-1 に，ユースケースのフレームワークを図 B-2 に示す．これらの図はマインドマップ作成ツールである FreeMind の ver.1.1.0 Beta 2 を使用して筆者が作成した．実験の際，被験者は同じバージョンの FreeMind を利用し，フレームワークを編集して BBS およびユースケースを作成した．フレームワークとして規定している編集不可のノードは，背景色を薄黄色とし他のノードと区別した．実験の手順を被験者に説明する際に，薄黄色のノードは規定のノードのため編集せず，白色のノードはツリー構造で自由に追加・編集して良いと説明した．複数のエージェントのユースケースを作成する場合は，ユースケースのフレームワークをファイルごとコピーして作成するように被験者に指示した．同一のエージェントのユースケースを複数作成する場合は，「ユースケース名」以下のノードをコピーして 1 ファイルに複数のユースケースを記載して構わないと被験者に説明した．

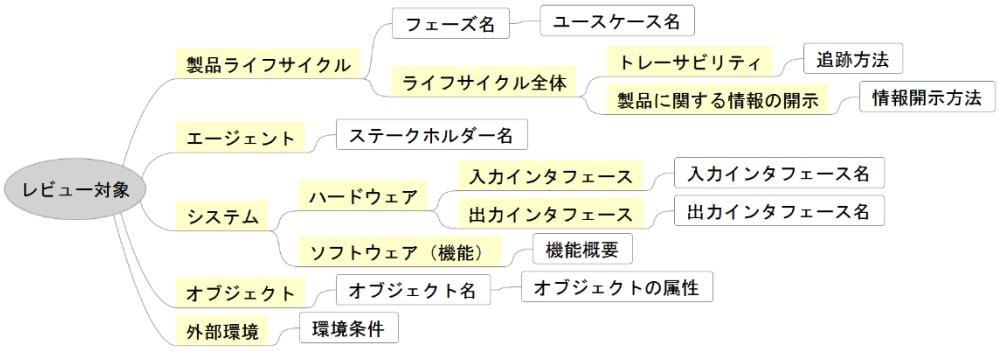


図 B-1　BBS のフレームワーク（実験用）

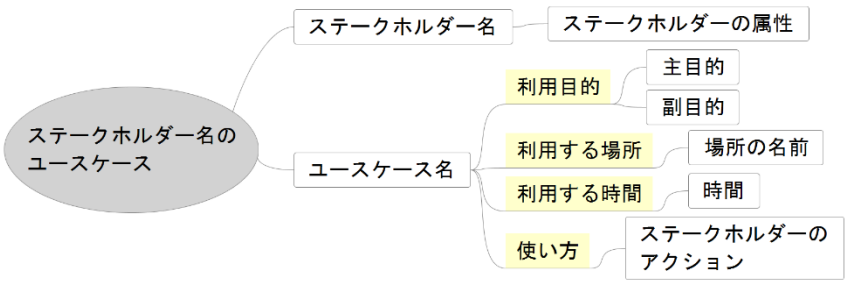
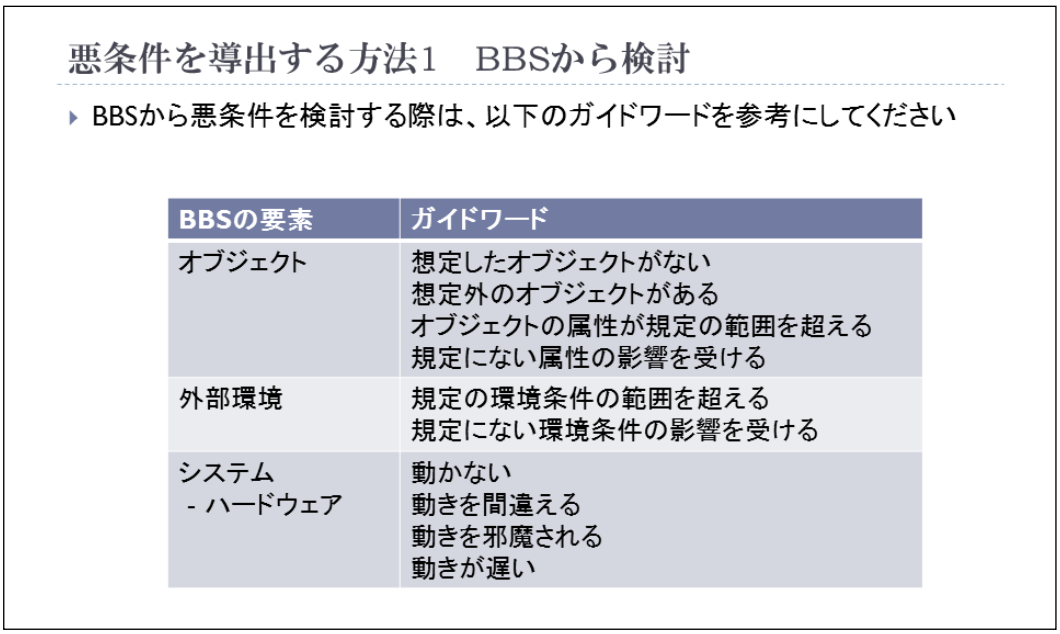


図 B-2　ユースケースのフレームワーク（実験用）

実験で使用した悪条件導出のガイドワードを図 B-3 と図 B-4，対策案のサンプルを図 B-5 に示す．実験の際，被験者にはこれらのガイドワードを印刷した資料を渡し，悪条件や対策案を検討する際に参考にしながら検討するように指示した．

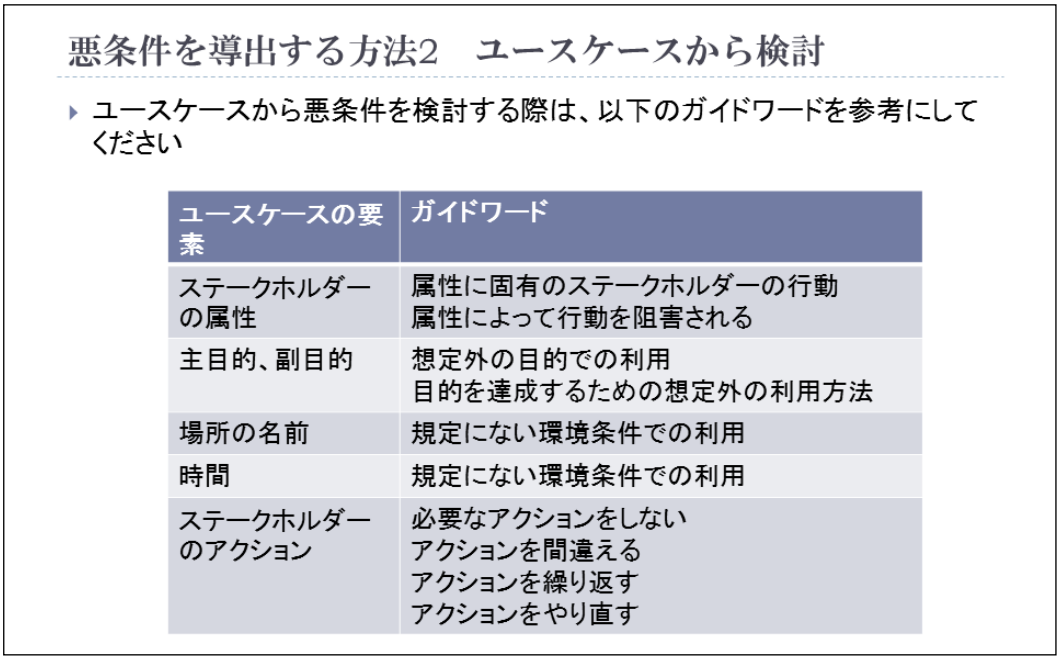


悪条件を導出する方法1　BBSから検討

- BBSから悪条件を検討する際は、以下のガイドワードを参考にしてください

BBSの要素	ガイドワード
オブジェクト	想定したオブジェクトがない 想定外のオブジェクトがある オブジェクトの属性が規定の範囲を超える 規定にない属性の影響を受ける
外部環境	規定の環境条件の範囲を超える 規定にない環境条件の影響を受ける
システム - ハードウェア	動かない 動きを間違える 動きを邪魔される 動きが遅い

図 B-3　BBS から悪条件を導出するガイドワード



悪条件を導出する方法2　ユースケースから検討

- ユースケースから悪条件を検討する際は、以下のガイドワードを参考にしてください

ユースケースの要素	ガイドワード
ステークホルダーの属性	属性に固有のステークホルダーの行動 属性によって行動を阻害される
主目的、副目的	想定外の目的での利用 目的を達成するための想定外の利用方法
場所の名前	規定にない環境条件での利用
時間	規定にない環境条件での利用
ステークホルダーのアクション	必要なアクションをしない アクションを間違える アクションを繰り返す アクションをやり直す

図 B-4　ユースケースから悪条件を導出するガイドワード

指摘を検討する方法

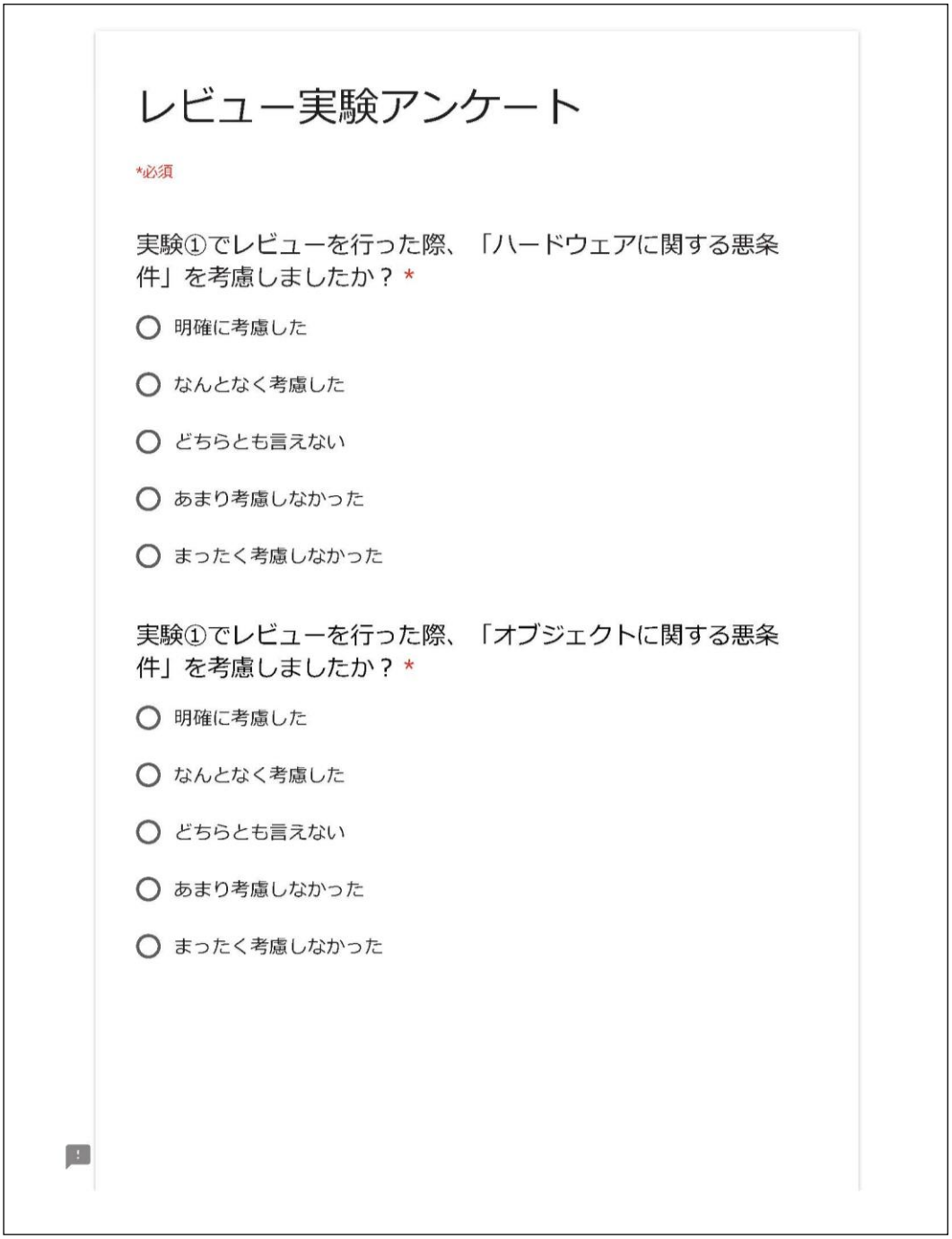
- 指摘を検討する際は、以下を参考にしてください
  - 1つの悪条件に対して複数の対策案を記載して構いません

対策のタイプ	対策のポイント	対策案のサンプル
悪条件の回避	ユーザ	悪条件が発生しないように、〇〇という手順をユーザに通知してはどうか
	システム	悪条件が発生しないように、システムが〇〇をできるようにしてはどうか
	外部環境	悪条件が発生しないように、外部環境に対して〇〇という対策を行ってはどうか
悪条件の保有	ユーザ	悪条件への対策がないことを事前にユーザに通知し、理解したうえで使ってもらってはどうか
	システム	-
	外部環境	-
悪条件の低減	ユーザ	悪条件が発生した場合の対策として、〇〇という手順をユーザに通知してはどうか
	システム	悪条件が発生した場合の対策として、システムが〇〇をできるようにしてはどうか
	外部環境	悪条件が発生した場合の対策として、外部環境に対して〇〇という対策を行ってはどうか
悪条件の移転	ユーザ	悪条件の影響を受けないように、ユーザが〇〇という別の方法を行ってはどうか
	システム	悪条件の影響を受けないように、〇〇という別の機能を作るか、別の製品・システムを使ってはどうか
	外部環境	-

図 B-5　対策案のサンプル

Appendix C　実験 1 のアンケートフォーム

実験 1 で使用したアンケートフォームを掲載する．被験者には Web ブラウザでアンケートフォームにアクセスしてもらい，回答を入力してもらった．アドホックレビューを「実験①」，提案手法によるレビューを「実験②」と記載している．



実験①でレビューを行った際、「外部環境に関する悪条件」を考慮しましたか？ *

- ○ 明確に考慮した
- ○ なんとなく考慮した
- ○ どちらとも言えない
- ○ あまり考慮しなかった
- ○ まったく考慮しなかった

実験①でレビューを行った際、「ユースケースに関する悪条件」を考慮しましたか？ *

- ○ 明確に考慮した
- ○ なんとなく考慮した
- ○ どちらとも言えない
- ○ あまり考慮しなかった
- ○ まったく考慮しなかった

実験①について難しかった点、ご意見や気づきがあれば自由に記載してください

回答を入力

実験②でレビューを行った際、「ハードウェアに関する悪条件」を考慮しましたか？ *

- ○ 明確に考慮した
- ○ なんとなく考慮した
- ○ どちらとも言えない
- ○ あまり考慮しなかった
- ○ まったく考慮しなかった

実験②でレビューを行った際、「オブジェクトに関する悪条件」を考慮しましたか？ *

- ○ 明確に考慮した
- ○ なんとなく考慮した
- ○ どちらとも言えない
- ○ あまり考慮しなかった
- ○ まったく考慮しなかった

実験②でレビューを行った際、「外部環境に関する悪条件」を考慮しましたか？ *

- ○ 明確に考慮した
- ○ なんとなく考慮した
- ○ どちらとも言えない
- ○ あまり考慮しなかった

○ まったく考慮しなかった

実験②でレビューを行った際、「ユースケースに関する悪条件」を考慮しましたか？ *

○ 明確に考慮した

○ なんとなく考慮した

○ どちらとも言えない

○ あまり考慮しなかった

○ まったく考慮しなかった

実験②について難しかった点、ご意見や気づきがあれば自由に記載してください

回答を入力

1/1 ページ

送信

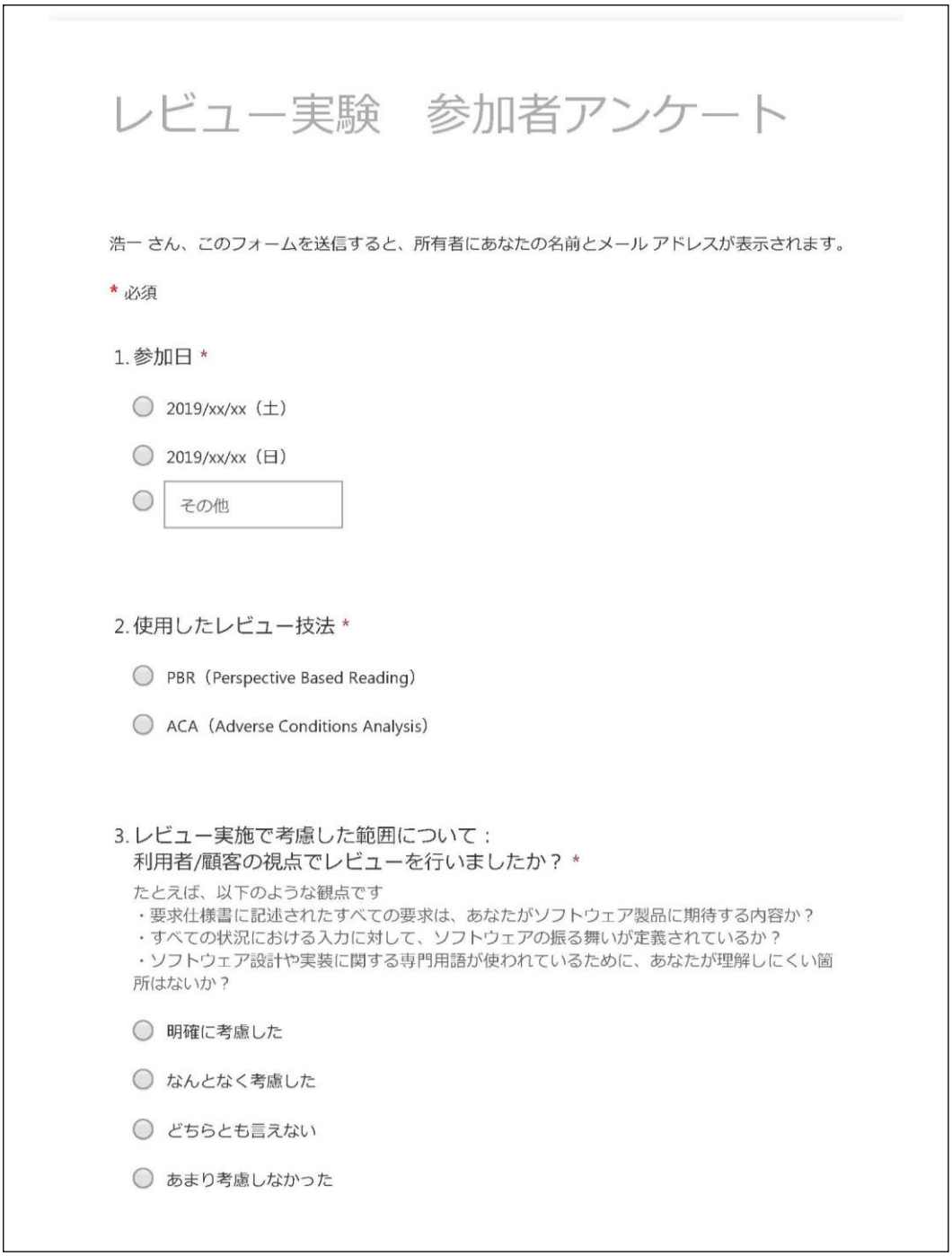
Google フォームでパスワードを送信しないでください。

このコンテンツは Google が作成または承認したものではありません。 不正行為の報告 - 利用規約 - プライバシーポリシー

Google フォーム

Appendix D　実験 2 のアンケートフォーム

実験 2 で使用したアンケートフォームを掲載する. 被験者には Web ブラウザでアンケートフォームにアクセスしてもらい, 回答を入力してもらった.



レビュー実験　参加者アンケート

浩一 さん、このフォームを送信すると、所有者にあなたの名前とメール アドレスが表示されます。

* 必須

1. 参加日 *

- 2019/xx/xx（土）
- 2019/xx/xx（日）
- その他

2. 使用したレビュー技法 *

- PBR（Perspective Based Reading）
- ACA（Adverse Conditions Analysis）

3. レビュー実施で考慮した範囲について：
利用者/顧客の視点でレビューを行いましたか？ *

たとえば、以下のような観点です
・要求仕様書に記述されたすべての要求は、あなたがソフトウェア製品に期待する内容か？
・すべての状況における入力に対して、ソフトウェアの振る舞いが定義されているか？
・ソフトウェア設計や実装に関する専門用語が使われているために、あなたが理解しにくい箇所はないか？

- 明確に考慮した
- なんとなく考慮した
- どちらとも言えない
- あまり考慮しなかった

○ まったく考慮しなかった

4. レビュー実施で考慮した範囲について：
設計者/開発者の視点でレビューを行いましたか？ *

たとえば、以下のような観点です
・設計に必要な情報（機能、性能、設計上の制約）はすべて定義されているか？
・すべての外部インタフェースは明確に定義されているか？
・要求仕様書の内部で、他の記述と互いに矛盾しているために、設計すべき内容が理解できない部分があるか？

○ 明確に考慮した

○ なんとなく考慮した

○ どちらとも言えない

○ あまり考慮しなかった

○ まったく考慮しなかった

5. レビュー実施で考慮した範囲について：
テスト担当者の視点でレビューを行いましたか？ *

たとえば、以下のような観点です
・要求される正確性および精度が定義されているか？
・すべての機能が要求を満たしていることを確認する方法について、あなたは明確に理解できているか？
・すべての機能が正しい値を出力していることを確認する方法について、あなたは明確に理解できているか？

○ 明確に考慮した

○ なんとなく考慮した

○ どちらとも言えない

○ あまり考慮しなかった

○ まったく考慮しなかった

6. レビュー実施で考慮した範囲について：
ハードウェアに関する悪条件を考慮してレビューを行いましたか？ *

たとえば、以下のような観点です
・ハードウェアが動かない

・ハードウェアが動きを間違える

- ○ 明確に考慮した
- ○ なんとなく考慮した
- ○ どちらとも言えない
- ○ あまり考慮しなかった
- ○ まったく考慮しなかった

7. レビュー実施で考慮した範囲について：
オブジェクトに関する悪条件を考慮してレビューを行いましたか？ *

※オブジェクト・・・レビュー対象製品に関連する物体
例えば、以下のような観点です
・想定したオブジェクトがない
・想定外のオブジェクトがある

- ○ 明確に考慮した
- ○ なんとなく考慮した
- ○ どちらとも言えない
- ○ あまり考慮しなかった
- ○ まったく考慮しなかった

8. レビュー実施で考慮した範囲について：
外部環境に関する悪条件を考慮してレビューを行いましたか？ *

例えば、以下の観点です
・規定の環境条件の範囲を超える
・規定にない環境条件の影響を受ける

- ○ 明確に考慮した
- ○ なんとなく考慮した
- ○ どちらとも言えない
- ○ あまり考慮しなかった
- ○ まったく考慮しなかった

9. レビュー実施で考慮した範囲について：
ユースケースに関する悪条件を考慮してレビューを行いましたか？ *

例えば、以下の観点です
・想定外の目的での利用
・規定にない環境条件での利用
・ユーザが必要な行動を行わない
・ユーザが行動を間違える

- ○ 明確に考慮した
- ○ なんとなく考慮した
- ○ どちらとも言えない
- ○ あまり考慮しなかった
- ○ まったく考慮しなかった

10. 本日使用したレビュー技法を業務で活用したいと思いますか？ *

現在、要求仕様書のレビューを担当している方は現在の業務で使うかお答えください。担当していない方は今後担当することになったと仮定して、お答えください。

- ○ 非常にそう思う
- ○ そう思う
- ○ どちらとも言えない
- ○ あまりそう思わない
- ○ 全くそう思うわない

11. 先ほどの質問に「非常にそう思う」「そう思う」と答えた方はその理由を選択してください（複数選択可） *

- ☐ 重要な指摘を出せそうだから
- ☐ レビューの時間を短縮できそうだから
- ☐ 普段思いつかない指摘を思いつくことができそうだから
- ☐ 指摘の漏れを減らせそうだから

- [ ] 指摘の途中経過を含めて顧客に説明しやすそうだから
- [ ] その他

12. 先ほどの質問に「どちらとも言えない」「あまりそう思わない」「全くそう思わない」と答えた方はその理由を選択してください（複数選択可） *

- [ ] 難易度が高いから
- [ ] 実施するのに時間がかかるから
- [ ] 効果があるか不明だから
- [ ] 技法がなくても同様の指摘を出せそうだから
- [ ] 自分の担当する製品にこの技法を適用するイメージが湧かないから
- [ ] 技法を理解できていないから
- [ ] その他

13. その他、質問やご意見がありましたら記載してください

回答を入力してください

- [ ] 自分の回答についての確認メールを受け取る

送信

このコンテンツはフォームの所有者が作成したものです。送信したデータはフォームの所有者に送信されます。パスワードを記載しないでください。

Powered by Microsoft Forms | プライバシーと Cookie | 利用規約