

## 研究コース2（設計着手前レビューチーム）

「要求には無いが想定しておくべき条件」に

### 着目した設計着手前レビューの提案

- 要求仕様の抜け漏れを防ぎ開発の前提条件のちゃぶ台返しによる

大幅な手戻りを防止-

Review before start designing focused on 'conditions not specified in requirements'  
- Prevent both missing requirements specification and major rework due to substantial change in development prerequisites -

研究員 : 村田 努 (オムロンヘルスケア株式会社)  
小川 聡 (株式会社 IHI)  
篠瀬 裕太郎 (株式会社リンクレア)  
主査 : 中谷 一樹 (TIS 株式会社)  
副主査 : 上田 裕之 (株式会社 DTS インサイト)  
アドバイザー : 安達 賢二 (株式会社 HBA)

#### 研究概要

発注者からの要求には無いが想定しておくべき条件が漏れていることが後工程で判明し、大きな手戻りとなることがある。このような事態に陥る原因の1つに、提示された要求をそのまま仕様に落とす、すなわち要求に対するレビューの適切な実施ができていないという問題がある。そこで我々は提示された要求に漏れがないかを検出する仕組みを考案した。実際の要求仕様書を用いた検証を実施した結果、その有効性が確認できた。提案手法と検証結果をここに報告する。

**Abstract** It is still hard to root out the unexpected specification change or addition, in latter development phase. One of the reasons is that developers may not be able to recognize leaks in case not all requirements are explicitly stated by customers. Since the customers are not software professional, it is difficult to expect customers prepare overall and perfect requirements. To achieve the project successful, developers should suppose or predict the necessary but leaked requirements. In this study, we propose the review method with framework for finding such concealed requirements, and describe effectiveness.

## 1. はじめに

### 1.1 背景

ソフトウェア開発において、発注者から提示された要求をもとに開発者はシステム（コンピューターで組み合わせられて機能しているハードウェアやソフトウェアの全体）を構築することが一般的であるが、要求にない項目が後工程で必要であることが判明し、その対応で大きな手戻りが発生する可能性がある。このような、要求にはないが開発者側でも容易に想定できる項目を洗い出し、仕様に落とすことができれば、欠陥流出防止につながる。

しかしながら、現場の多くは、発注者から提示された要求のみをそのまま仕様に落とすことに注力しており、要求を精査するための適切なレビューを実施できていないのが現状である。

## 1.2 本論文の前提となる事項

### ＜システムの開発形態＞

本論文では、システムの開発形態として、「要件定義・仕様定義を行う人と、設計以降を担当する人が別の場合(例えば、設計以降の開発に参画する開発請負)」を前提としている。本論文で想定したシステムの開発形態の例を図 1-1 に示す。

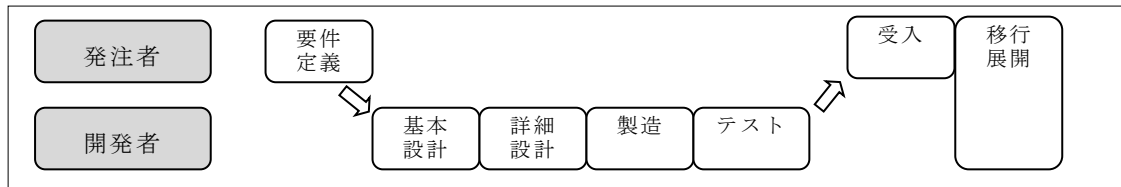


図 1-1. システムの開発形態の例

「発注者」… 要求を出す人, または要件定義・仕様定義を行う人

「開発者」… 要求または定義された要件・仕様を元にシステムを開発する人

## 2. 解決すべき問題

### 2.1 現状分析

発注者は開発のプロでは無いため、発注者が気付かなかった部分は要求として示せていないことがある。このような時でも、開発者が抜け漏れている仕様に気づき補完することが出来れば、手戻りを防ぐことができる。しかし、発注者から明示的に提示された要求に対して開発者が忠実に実現していくと、要求に現れてきていない項目は見落とされたまま開発が進み、結果的に後工程で手戻りが発生してしまうことがある。

1.1 章で取り上げた問題「要求にない項目が後工程で必要であることが判明し、その対応で大きな手戻りが発生する」に関して、ソフトウェア開発の現場における実態を把握するために、リリース完了済みの 1 つの開発プロジェクトの障害データについてその原因を分析した。なお、発注者からの要求に含まれていたのか否かを明らかにするため、本論文では、システムの仕様を表 2-1 のように分類している。

文献<sup>[1][2]</sup>では、「意図する使用(intended use)」という用語が用いられているが、本論文ではソフトウェアの実現方法も含めて「意図する仕様」という用語を用いる。文献<sup>[1][2]</sup>では、事故発生時でも「合理的に予見不可能」と認定される場合には免責され得る分類の記載があるが、本論文では着手前のレビュー時に適用することから、予見できたもののうち発注者との間で異常・不適切な使われ方として合意し得るものを仕様分類 C とした。

表 2-1. システムの仕様分類

仕様分類	説明
A	意図する仕様 発注者から提示された要求に基づいた仕様。
B	提示されていないが合理的に予見可能な仕様 発注者からの要求には無いが、利用される場面を想像すれば、推測し得る予見可能な仕様。
C	予見できた仕様のうち本来の目的から逸脱した異常・不適切な仕様 無謀な使用、(製品/システムに対する)妨害行為、または製品/システムの使い方を意図的に無視した仕様および使用方法。

障害データの分析結果を図 2-1 に示す。設計不良 19 件のうち、抜け漏れが原因となっているものが 13 件あった。そのうち仕様分類 B の漏れが原因のものが計 8 件であった。ソフトウェア開発の現場では、発注者からの要求に対して精査が適切に出来ておらず、大きな手戻りとなっていることがこの分析結果からも分かった。

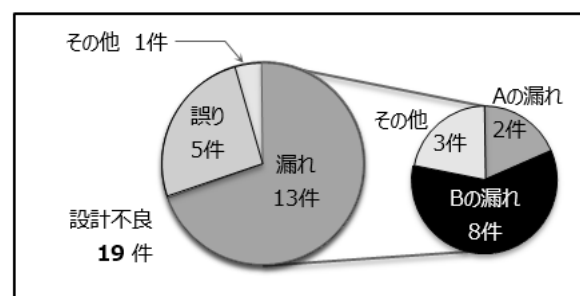


図 2-1. 障害データの分析結果

## 2.2 課題提起

発注者は、実現したいこと（仕様分類 A）に高い関心を持つ。一方、開発者は本来、要求に無くても利用される場面を想像すれば容易に想定できる仕様（仕様分類 B）も含めて仕様に落とし込む必要がある。しかし開発者は、開発の早期着手への想いが先行し、発注者からの要求に対する精査が不十分なまま開発を進めてしまうことがある。その結果、仕様分類 A および仕様分類 B の洗い出しが不完全となる。また仕様分類 B に位置付けるべきものを予見できているにも関わらず、本来の目的から逸脱した異常・不適切な仕様（仕様分類 C）と決め付ける、あるいは排除してしまうこともある。そしてその認識が関係者間で合意形成できていないまま開発を進めてしまい後工程で要求の漏れが判明した場合、大きな手戻りとなってしまふことがある。

このような問題を解決するにあたり、先行研究を調査したが、適切な先行研究を見つげ出すことができなかった。そこで今回我々で、発注者からの要求に対して、仕様分類 B に着目して実施する設計着手前レビューを考案することにした。新しいレビュー手法を考案するにあたり、以下の RQ を設定する。

**RQ:** 提案レビュー手法により、仕様分類 B にあたる要求仕様の漏れを検出できるか。

## 2.3 先行研究

課題を解決するにあたり、「発注者からの要求を精査する」、「合理的に予見可能な仕様」の2つの観点で先行研究を調査した。その結果を以下に示す。

### (1) 発注者からの要求を精査

「発注者からの要求を精査」に関して、コンセプトベースドレビューの提案が、我々が実現したいことと一致しており、参考になると考えた。

表 2-2. 「発注者からの要求を精査」に関連する先行研究

先行研究	特徴	備考
コンセプトベースドレビュー <sup>[3]</sup>	レビューを実施する前に、市場のニーズや開発の背景からレビューの観点を導き出し、その観点をもとにレビューする。	ユーザによるシステム利用時の課題や解決したい問題に応える仕様となるような指摘を行える。

### (2) 合理的に予見可能な仕様

「合理的に予見可能な仕様」に関連する先行研究として、ISO/IEC Guide51<sup>[2]</sup>というリスクマネジメントガイドがあった。本論文では、このガイドで示されているシステムの使い方の分類を参考にして、表 2-1 に示した通り、システムの仕様分類を定義した。仕様分類ごとの具体的な導出方法については我々で考えることにした。なお、コンセプトベースドレビューの提案では、カスタマージャーニーマップを用いてシステムの使われ方を導出していた<sup>[3]</sup>。システムが使われる場面に焦点を絞り、仕様分類 A が中心になっているため、対象範囲やアプローチが異なるものであった。

表 2-3. 「合理的に予見可能な仕様」に関連する先行研究

先行研究	特徴	備考
ISO/IEC Guide51 (リスクマネジメントガイド) <sup>[2]</sup>	「意図する使用」「合理的に予見可能な誤使用」「合理的に予見不可能な誤使用または異常使用」の3つに分類している。	・本提案手法におけるベースとなる考え方
ペルソナ手法 (UX) <sup>[4][5]</sup>	5W1H の「who」「what」「why」の要素を中心に UX 観点を膨らませていく手法	・表 2-1 の仕様分類だと「仕様分類 A」相当が中心 ・狩野モデルの「魅力的品質」 <sup>[6]</sup> を対象 ・十分に経験を積んだ有識者が必要 <sup>[4]</sup>
コンセプトベースドレビュー <sup>[3]</sup>	システムが使われる場面に焦点を絞っている。	・表 2-1 の仕様分類だと「仕様分類 A」相当が中心 ・狩野モデルの「魅力的品質」 <sup>[6]</sup> を対象
ビジネスリスクに着目したレビュー <sup>[7]</sup>	リスク欠陥ツリー (RDT) を用いて、ビジネスリスク (BR) に直結するシステムリスク (SR) からレビューポイントを導く枠組み	・リスクについて検討するという点で、本提案手法と似ている部分がある

### 3. 解決策の提案

先行研究を調査したが、我々が解決したいことに直接応えてくれるものを見つけることができなかつたため、「ISO/IEC Guide51」のシステムの使い方の分類、「コンセプトベースドレビュー」の考え方、及び「ペルソナ手法」の属性の考え方を参考にしつつ、我々は、仕様分類 B に着目し、それを設計着手前に実施するレビュー手法を独自に考案した。

我々が考案した手法について、以下に示す。

#### (1) 本手法の名称

『「要求には無いが想定しておくべき条件」に着目した設計着手前レビュー』

NLR 法 (suppose and make up for Necessary but Leaked Requirements)

#### (2) 本手法のコンセプト

本手法は、以下のコンセプトとする。

- 設計着手前に発注者からの要求に抜け漏れが無いが精査する。
- 仕様分類 B (提示されていないが合理的に予見可能な仕様) に着目する。
- システムの利用時だけでなく、開発時から廃棄/廃止されるまでに関与するステークホルダーの役割や行動をライフサイクルの各段階で意識する。

#### (3) 使用するもの

本手法では、以下を使用する。

INPUT : 上流工程の成果物 (要求仕様書など発注者からの要求を記載した文書)

TOOL : ① 着手前要求確認フレームワーク (表 3-1)

② 「区分・パラメータ」シート (図 3-1)

表 3-1. 着手前要求確認フレームワーク (例として「勤怠管理システム」を記入)

No.	5W1H	観点	①INPUTを確認した結果/ ②抜け漏れとして想定すべきこと	使用分類			備考/コメント (関係者と合意したこと、理由などを記載)	
				INPUT	レビュー時	合意		
1	What	何をやる製品/サービスか?	社員の勤怠管理				・1次リリースで対象外とするものは、分類Cとする。 ・2次リリース向けでは、本フレームワークを使用して改めて分析し直す。	
2	Who	誰(どのシステム)が使う(使われる)か? → 区分・パラメータ シート参照	1 社員 2 開発担当者 3 社員(人事担当者)	A		A		
3	When	いつ利用する(利用される)か? → 区分・パラメータ シート参照	1 出勤時 2 退勤時	A		A		
4	Where	どこで使う(使われる)か? / どんな状況で使う(使われ)る? → 区分・パラメータ シート参照	1 オフィス内 2 オフィス外(国内) 3 オフィス外(海外)	A		A		
5	How	どのように使う(使われる)か?	ユースケースを列挙する (インプット資料から読み取れるもの、 レビューで指摘したもの)					
			ID (5.1) ライフサイクル 5.2) 誰が(どのシステム)が # (5.2) どうする					
		A 開発中	開発担当者	が	開発を行う		A	A
		B テスト	開発担当者	が	テストを行う		A	A
		C リリース	開発担当者	が	リリースを行う		A	A
		D デプロイ	開発担当者	が	デプロイを行う		A	A
		E1 使用(通常自)	社員	が	出勤時に出勤時間を入力する	A		A
					退勤時に退勤時間を入力する	A		A
			社員	が	自分以外の出退勤時間を入力する		B	B
			社員	が	出張・不在の為に期日を過ぎてから出退勤時間を入力する		B	C
			社員	が	海外からVPNを使い、出退勤時間を入力する		B	C
			社員	が	勤怠表を確認する	A		A
			社員	が	勤怠表の承認を行う		B	C
			社員(人事担当者)	が	全社員の勤怠時間の確認を行う		B	C
E2 使用(災害時)								
F 運用								
G 移行	開発担当者	が	移行		A	A		
H メンテナンス	開発担当者	が	定期バックアップを行う		B	C		
J バージョンアップ								
K 運用終了(廃止)								
6	Why	何のために製品/サービスを作るのか? この製品/サービスは、お客様にどのような価値やアトカムを提供するのか?	紙での勤怠管理をやめて、システム化を行い、全社員が勤怠システムから入力できるようにする。会社が労働基準監督署から指導を受けることがないように、法定時間を超えた労働をさせないようにして、社員の健康管理を把握できるようにする。					

Who	探査 No.	区分 (Segment)	パラメータ (Parameter)	
	○	1	対象者	購入者, 所有者, 開発担当, 生産担当, 品質担当, …
When	探査 No.	区分 (Segment)	パラメータ (Parameter)	
		1	頻度	毎日, 週に1度, 月に1度, …
Where 【場所】	探査 No.	区分 (Segment)	パラメータ (Parameter)	
		1	国や地域	日本, 北米, 欧州, …
Where 【状況】	探査 No.	区分 (Segment)	パラメータ (Parameter)	
		1	温度	0度未満, 0~40度, 40度超え
		2	湿度	30%RH未満, 30~95%RH, 95%RH超え
		3	姿勢	座っている, 立っている, 寝ている, …
		4	残業状況	無し, ~10時間以内, ~40時間以内, …
		5	習熟度合	指導者レベル, 中級レベル, 初心者レベル, 未経験, …
		6	作業場所の雰囲気	和やか, 殺伐, ミスが許されない, …
	:	:	:	

図 3-1. 「区分・パラメータ」シート

「区分・パラメータ」シートは、着手前要求確認フレームワーク(表 3-1)の Who, When, Where に関する抜け漏れを見つけるためのツールとして、予めいくつかのパターンを掲載している。これはペルソナ手法における属性の考え方を応用したものである。例えば要求仕様に「オペレータ」と記載されている場合、知識のある人が操作するものと暗黙のうちに想定して読み進める可能性がある。しかし、このシートを活用すれば、初心者や代理で操作する人など、思い込みには陥らないよう気付きを得ることが出来る。更に、このシートは実際の業務やアプリケーションに合わせて加筆修正していくことで、ノウハウを形式知化し、経験の少ないレビューアでも想定すべき範囲を広げることにも役立てることが出来る。

#### (4) 実施手順

本手法の実施フローを以下に示す。

実施フローは「準備」「実施」「合意」の3つからなる。「準備」では、上流工程の成果物を入力する。「実施」では、TOOL である着手前要求確認フレームワークを使用して要求仕様の抜け漏れを推測・補完する。「合意」では、「実施」で記入した着手前要求確認フレームワークを使用して、発注者など関係者との合意形成を図る。

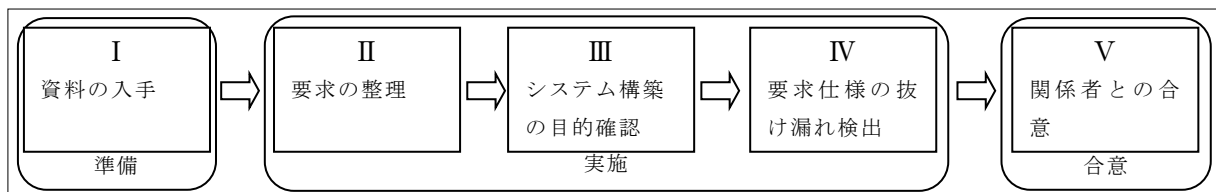


図 3-2. NLR 法の実施フロー

実施手順は以下の通りである。(更に詳細な手順は付録 3 に記載)

#### I. 資料の入手

上流工程の成果物(要求仕様書など発注者からの要求を記載した文書)を入力する。

#### II. 要求の整理

上流工程の成果物に記載されている要求をフレームワークに当てはめて整理する。ここで整理される項目は、原則全て仕様分類 A となる。

1. What 欄に、何をやる製品/システムのことがかかれているのかを記入する。
2. Who 欄に、誰(どのシステム)が使うのかを記入する。
3. When 欄に、いつ使う(使われる)のかを記入する。
4. Where 欄に、どこで、どのような状況で使う(使われる)のかを記入する。
5. How 欄の「ライフサイクル」「誰が」「どうする」を時系列順に書き出す。

#### III. システム構築の目的確認

6. Why 欄に、製品/システムが提供する価値(value)やアウトカム(outcome)が何かを推測して記入する。何のために構築するのか理解すると、抜けている観点に気づく可能性が高まる。

No.	5W1H	観点	①INPUTを確認した結果/ ②抜け漏れとして想定すべきこと	仕様分類			備考/コメント	
				INPUT	レビュー時	合意		
1	What	何をやる製品/サービスか?	BLE通信機能付き IoT懐中電灯					
2	Who	誰が使うか?	使用者 (身体的特徴は問わない)	A				
3	When	いつ利用するか?	停電時	A				
4	Where	どこで使うか? / どんな状況で使うか?	屋内、および屋外	A				
5	How	どのように使うか?						
		ID	ライフサイクル	誰が	どうする			
		H	操作	使用者 が	懐中電灯とスマホのペアリングを行う。	A		
J	使用	使用者 が	電源スイッチを押して、ライトを点ける。	A				
6	Why	何のために製品/サービスを作るのか?	突然の地震や停電の際には手元に懐中電灯が無くても、IoT技術を使い、スマートフォンの遠隔操作で明りを灯せる仕組みを提供することで、お客様の不安や緊張を緩和する。					

図 3-3. 要求の整理のイメージ図 (例として「IoT 懐中電灯」を記入)

#### IV. 要求仕様の抜け漏れ検出

以下の作業は集合レビューとして実施することでより効果が期待できるが、個人レビューでも手順に沿うことで効果を得ることができる。

- 過去に同様の製品/システムを開発している場合はその時の不具合/トラブルを元に、IIの2~5の該当箇所に追記する。同等でなくても、類似または応用できそうな製品/システムを開発している場合も同様に追記する。
- 上流工程の成果物に記載されている内容に対して、過去の不具合事例で抜け漏れないか確認する。
- 「誰が」「いつ」「どこで」「どのような状況で」を中心に抜け漏れないかを、集合レビューの場で確認して追記する。なお、図 3-1 に示した「区分・パラメータ」シートを活用することにより、抜け漏れや、発注者と開発者との間で認識齟齬が発生しやすい項目を効率的に洗い出すことができる。
- 「誰が」「いつ」「どこで」「どのような状況で」の観点を深掘りした後、同様にライフサイクルの段階ごとに「誰が」「どうする」として抜け漏れないかを、集合レビューの場で確認して追記する。発注者の想いに応えることばかりに目が行くと、保守など運用上必要な要件の部分が抜けやすくなるため、ライフサイクルを意識することが抜け漏れの発見に役立つ。

手順IVで追記される項目は、仕様分類 A, B, C の全てが対象である。

No.	5W1H	観点	①INPUTを確認した結果/ ②抜け漏れとして想定すべきこと	仕様分類			備考/コメント	
				INPUT	レビュー時	合意		
1	What	何をやる製品/サービスか?	BLE通信機能付き IoT懐中電灯					
2	Who	誰が使うか?	使用者 (身体的特徴は問わない)	A				
			開発担当者		A			
3	When	いつ利用するか?	停電時	A				
			常時 (点けっぱなし)		B			
4	Where	どこで使うか? / どんな状況で使うか?	屋内、および屋外	A				
			海外 (電波法が厳しい国)		B			
5	How	どのように使うか?						
		ID	ライフサイクル	誰が	どうする			
		A	開発中	開発担当者が	デバッグモードで変数値を確認する。		B	
		H	操作	使用者が	懐中電灯とスマホのペアリングを行う。	A		
		J	使用	使用者が	電源スイッチを押して、ライトを点ける。	A		
K	使用終了	介助犬が	電源スイッチを押して、ライトを消す。		B			
6	Why	何のために製品/サービスを作るのか?	突然の地震や停電の際には手元に懐中電灯が無くても、IoT技術を使い、スマートフォンの遠隔操作で明りを灯せる仕組みを提供することで、お客様の不安や緊張を緩和する。					

図 3-4. 要求仕様の抜け漏れ検出のイメージ図

#### V. 関係者との合意

抜け漏れの項目確認を行い、仕様分類 A, B, C の認識を合わせる。分類を変更した理由や、仕様・実現時期など双方で合意した内容を記録しておくことで、見積り範囲 (開発範囲) の明確化や、後日のトラブル防止にも役立つ。

## 4. 解決策の評価

### 4.1 評価方法

以下に評価手順を示す。

- (1) 実プロジェクトの要求仕様書（要件定義書）を使用して手順Ⅱ～Ⅳを実施する。
- (2) 設計着手前レビューを実施しない場合と、NLR法を適用した設計着手前レビューを実施した場合で要求仕様の項目数を比較する。
- (3) 被験者は、研究員メンバー2名と研究員以外のメンバー1名とする。
- (4) レビューの目的は、要求を精査して抜け漏れを検出することである。
- (5) 集合レビューではなく個人レビューで実施する。
- (6) 被験者にアンケートを実施して本手法を評価すると共に改善点を見つける。

### 4.2 評価結果

実験対象とした要求仕様書の特性を表4-1に、実験結果を表4-2に示す。

表4-1. 要求仕様書の特性

対象	業種	システムの種類	請負工程	現在の工程	レビュー対象	页数
X	製造業	社内業務支援	基本設計～テスト	基本設計着手前	要求仕様書	6
Y	サービス業	勤怠管理	要件定義～テスト	提案書作成中	提案依頼書	6
Z	製造業	社内業務支援	運用・保守	運用手順作成中	操作手順書	2

表4-2. 実験結果

実験対象	要求仕様の項目の重要度	設計着手前レビュー未実施			NLR法を適用した設計着手前レビュー実施			手順Ⅱ～Ⅳの実施時間(時間)
		要求仕様の項目数(件)			要求仕様の項目数(件)			
		仕様分類A	仕様分類B	仕様分類C	仕様分類A	仕様分類B	仕様分類C	
X	高	59	0	0	65	0	0	2
	中～低	0	0	0	0	3	4	
Y	高	55	0	0	56	4	0	8
	中～低	0	0	0	4	0	0	
Z	高	4	0	0	5	3	0	0.5
	中～低	0	0	0	0	0	0	

NLR法を適用した設計着手前レビューを実施した結果、設計着手前レビューを実施しない場合に比べて、要求仕様の項目を多く導き出すことができた。すなわちNLR法の適用により、要求の抜け漏れを検出することが出来た。

### 4.3 アンケート結果

被験者のアンケート結果を以下に示す。

#### (1) アンケート結果

被験者3名にアンケートを実施した結果、フレームワークの使いやすさ、抜け漏れ防止の効果、共に高評価であった。アンケートの結果を表4-3に示す。

表4-3. アンケート結果

質問	4. そう思う	3. 少しそう思う	2. あまりそう思わない	1. 思わない
1. フレームワークは使いやすいですか？	3	0	0	0
2. 要求の抜け漏れ防止に役立ちそうですか？	1	2	0	0

#### (2) NLR法に関する意見・要望

被験者からは、以下のような意見・要望が挙げられた。

- ・ 初見で見た要求仕様書であったが、フレームワークを使って5W1Hで纏めるため、細かく仕様を追いやすく、抜け漏れが洗い出しやすくなっている。
- ・ フレームワークの使用法に記載されている文章が多く読むのに時間がかかる。もっと簡潔に記載されていると更に効率的に実施できると思った。  
→ 実験後、本意見を踏まえて表示/非表示の機能をフレームワークに実装済み。
- ・ 要求仕様書そのものの品質が低い場合、仕様分類A相当の漏れが多く検出され、狙いとは異なるが、より重大な漏れが見つかるという点で効果があると感じた。
- ・ NLR法を継続して適用し、「区分・パラメータ」シートの内容やフレームワークのガイドラインを自分達で充実させていくことで有識者の知見を共有できる。

#### 4.4 結果の考察

2.2章で設定したRQに対し、実験結果を踏まえて考察を示す。

**RQ:** 提案レビュー手法により、仕様分類Bにあたる要求仕様の漏れを検出できるか。

実験結果から、仕様分類Bにあたる要求仕様の漏れを検出できるだけでなく、仕様分類A, B, C全ての領域で、重要度が中～低の要求仕様の漏れも多く検出できた。従ってNLR法は、仕様分類Bにあたる要求仕様の漏れを検出するのに有効であるといえる。また、集合レビューの中で、仕様分類Bにあたる要求仕様の漏れを検出するのが本来のNLR法の手順であるが、個人レビューでも検出できることが分かった。

#### 5. まとめ

##### 5.1 成果

発注者からの要求には無いが想定しておくべき条件が漏れていることが後工程で判明し、大きな手戻りとなるという課題に対して、その解決策として我々は、『「要求には無いが想定しておくべき条件」に着目した設計着手前レビュー』を考案した。

発注者の要求が明確な仕様分類A「意図する仕様」に開発者も注力してしまうのではなく、仕様分類B「提示されていないが合理的に予見可能な仕様」に焦点を当てることにより、これまで開発の後工程で発覚して仕様分類C（異常仕様）に意図的に分類していたものが、着手前に仕様分類Bに相当する領域まで想定範囲を広げることで、要求の抜け漏れを防ぐ（「想定外」を「想定範囲内」にする）ことが可能となる手法である。

実際のプロジェクトの要求仕様書を用いた検証実験を実施したところ、本手法を適用した場合、仕様分類Bの要求の抜け漏れを検出できることが確認できた。また被験者のアンケートでは考案したフレームワークの使用性・有効性について高評価が得られたことから、本手法は開発現場の人にとって大いに役に立つことが期待できる。

##### 5.2 今後の進め方

今後、本手法を実際のプロジェクトに多く適用していきたいと考えている。ただし以下の課題についても解決していかなければならないと考えている。

本手法の手順の中で、要求にはない仕様を想起していく際に用いる「区分・パラメータ」シートにより、Who, When, Whereという切り口から機能要件の抜け漏れの気付きは得やすくなると考えられるが、非機能要件までカバーしているとは言い難い。ソフトウェアの品質特性（ISO/IEC 9126及びISO/IEC 25010）で挙げられている各特性なども切り口や区分として取り入れ、非機能要件を含めた抜け漏れにも気付ける手法に昇華させていきたい。

#### 参考文献

- [1] 経済産業省, 「リスクアセスメントハンドブック実務編」, 2011
- [2] ISO/IEC Guide 51 “Safety aspects - Guidelines for their inclusion in Standards”, 2014
- [3] 小楠 聡美, 「顧客の期待に応えるためのコンセプトベースドレビューの提案」, SQiPシンポジウム2018
- [4] 益子 なるみ, 森脇 正寿, 柚木崎 由貴, 村上 仁, 金山 豊浩, 三井 英樹, 村上 和治「要求段階で齟齬を減らすための合意形成手法の提案」, SQiP研究会第33年度(2017年度), 研究コース3
- [5] 川西 裕幸, 栗山 進, 潮田 浩, 「UXデザイン入門」, 日経BP社, 2012
- [6] 狩野 紀昭, 瀬楽 信彦, 高橋 文夫, 辻 新一, 「魅力的品質と当り前品質」, 品質, Vol. 14, No. 2, pp147-156, 1984
- [7] 小田切 勇人, 近藤 忍, 藤崎 祐美子, 細川 宣啓, 永田 敦, 藤原 雅明, 森崎 修司, 「ビジネスリスクに直結するレビューポイント導出方法の提案」, SQiP研究会第28年度(2012年度), 研究コース3(海グループ)