

# 「観点 60」-仕様書の問題点指摘手法

田村 正吾、内藤 駿樹、藤田 真広、宮崎 行規、森田 文弥<sup>A)</sup>、山本 修一郎<sup>B)</sup>

<sup>A)</sup> 株式会社ベリサーブ

<sup>B)</sup> 名古屋大学

## Viewpoint 60 – Method to point out the problems of specification

Syougo Tamura, Toshiki Naitou, Masahiro Fujita, Fumiya Morita, Shuichiro Yamamoto

Veriserve

Nagoya University

### 概要

「観点 60」とは、仕様書の問題点を指摘した実例を抽象化し、標準化した観点階層図を活用することにより、仕様書の問題点に対する指摘漏れを減らすための手法である。本稿では、経験別に「観点 60」使用の有無によって、仕様書の問題点の指摘数が増えるかどうかの実験を行った結果を明らかにする。

### Abstract

Viewpoint60 is a method to reduce leakages of specification problems extraction by standardized guidewords designed by the abstraction based on experiences of specification problems cases. In this paper, the Viewpoint60 is proposed and an experimental result is also shown by comparing the number of specification problems by the use or non-use of the Viewpoint60.

## 1.はじめに

ソフトウェア開発に当たり、開発者の不注意や気づかぬうちに作りこまれるバグの生存期間を短縮することは、ソフトウェア開発の効率化を求める上で、必要不可欠である。

ソフトウェア評価は開発段階の潜在的なバグを検出し、修正させることに注目しているが、開発段階でバグを作りこませない仕様書・設計書の作成も課題となっている。さらにソフトウェア開発においても、属人化や分散された技術の標準化は、コストダウンを図る上で常に付きまとう。

バグを作りこませない仕様書・設計書の作成も例外ではなく、今後はこの技術を標準化することで、ソフトウェア開発におけるコストダウンを図る必要が出てくる。我々の各現場では、バグを作りこませない設計・製造をさせるための活動が既に取り組みされており、その技術を標準化するための研究を行っている。

その研究から、バグを作りこませない仕様書の作成は、共通する観点があり、それらを分類・抽象化することで、標準化できることがわかってきた。本稿では、分類・抽象化した観点の活用と、その有効性について考察する。

## 2 仕様書レビューの課題

以下で述べている「仕様書レビュー」とは、ソフトウェア開発を行うにあたっての要求仕様書や機能仕様書に対して、行うレビュー全般のことを指している。

仕様書レビューの課題として、経験の浅いメンバは指摘が少ないという点が上げられる。

経験の豊富なメンバと比較をした際に、指摘が少ないということは指摘の漏れ、見落としがあると考えられる。

この問題は、“指摘をするための技術や知識などの経験の差”であることが原因であると仮説を立てた。

指摘を多くするメンバは日々の業務の積み重ねにより得た技術・知識を活用し、仕様書レビューを行っている。

一方で、経験の浅いメンバはその技術・知識が養われていないため、技術・知識を活用した仕様書レビューを行うことができない。結果として指摘漏れや見落としといった不完全な成果につながってしまうことがある。

この課題を解決するためには、仕様書レビューでの指摘事項を発見するための技術や知識を「観点」としてまとめる必要がある。「観点」をまとめた観点階層図を用いれば、経験の浅いメンバと経験豊富なメンバと

の指摘数の差異を減少させることができ、同程度の成果をあげることができると考えた。

また、多数の観点を列挙するだけでは理解が困難である。このため、観点を階層化して整理する必要がある。

### 3.「観点 60」とは

上述した課題を解決するために、階層化して整理した観点を「観点 60」と呼ぶ。

「観点 60」を作成するために、ソフトウェア評価の業務経験が3年から6年未満のメンバからなるプロジェクトを立ち上げた。

また、メンバ数5名程で取り組んでいる。

以下、「観点 60」を作成するにあたってのプロジェクトの推進をしたメンバを「推進メンバ」、「観点 60」を作成するにあたって実証実験などに協力したメンバを「協力メンバ」とする。

「観点 60」を作成するにあたり、推進メンバが協力メンバ数名へ各々が持っている観点についてヒヤリングを実施した。

協力メンバは各々が固有に持っている観点を活用して仕様書レビューを行っている。すなわち個人が持つ独自のノウハウで仕様書の指摘を行っている。そのため感覚による部分が大きく、このままでは共有できる手段がない。

まずは各々が仕様書の問題を指摘する際に、どういった箇所に主眼を置いているかなどをヒヤリングし、感覚として行っている部分を整理した。

収集した観点をカテゴリ化した結果、「入力」、「出力」、「処理」、「環境の状態」にまとめられた。そのカテゴリからどのような指摘ができるか示した一覧を図1に示す。

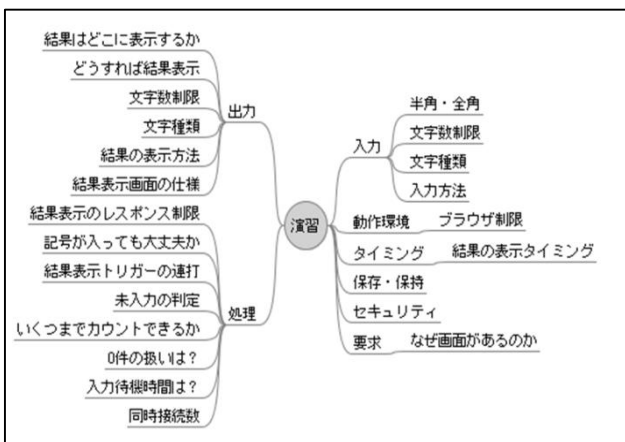


図 1 収集した観点のカテゴリ化結果

図 1は親ノードにカテゴリ化した観点の「入力」、「出力」、「処理」、「環境の状態」から発見することができる指摘を子ノードとし、階層的に並べた。

次に、実際に仕様書を指摘した際の事例を収集し、その事例を発見するためにはどのような観点が必要なのかを協議し、観点の精査・整理を行った。

そして精査・整理した観点を元に、より詳細で実用性の高い観点階層図を作成した。

これが「観点 60」である。(図 2)

このとき、まずは現実的に収集可能な 60 個の事例を収集することを目標とし取り組んだため、「観点 60」と名づけた。

結果として、事例は 60 個以上集まっている。事例や観点の数が 60 個であるから「観点 60」と命名したわけではない。

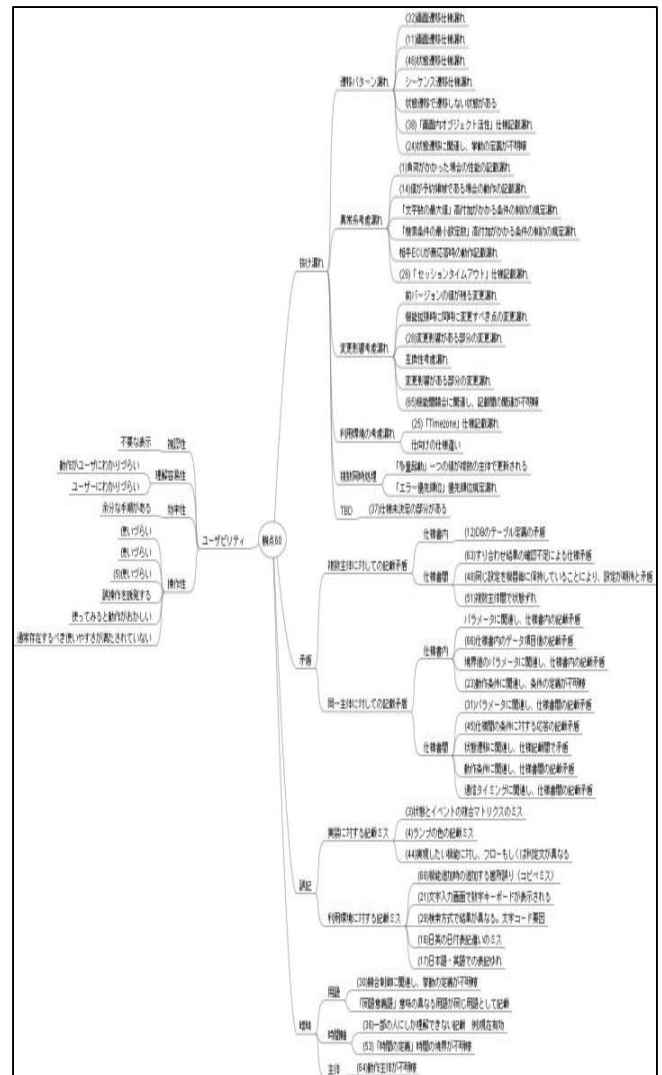


図 2「観点 60」

「観点 60」は大元となる親ノードを「抜け漏れ」、「矛盾」、「誤記」、「曖昧」、「ユーザビリティ」の5つにカテゴリ化した。その後、収集した事例を子ノードに並べた。

しかし、情報を整理することを目的としているマインドマップ形式の表だけでは、実際に使用するシーンを想定した際に、実用的ではないということが予想された。

そこで、どのシーンでも使える実用性の高い形を考慮し、表計算ソフト形式に変換を行った。(図 3)

				どこを			仕様仕様書 事例%
				入力	処理	出力	
仕様仕様書事例から抽出した観点 どんな観点で	A ユーザビリティ	A-1	視認性				
		A-2	理解容易性				
		A-3	操作性				
		A-4	操作性				5
	B 抜け漏れ	B-1	遷移(ボタン)漏れ				32,11,46,30,24
		B-2	異常系考慮漏れ				1,14,26
		B-3	デザイン変更による影響考慮漏れ				29
		B-4	システム変更による影響考慮漏れ				B-3D,親生
		B-5	利用環境の考慮漏れ				25
		B-6	複数同時発生の考慮漏れ				
		B-7	「E0」(仕様未決定) 実施の可否(仕様未決定)				37
	C 矛盾	C-1	記載矛盾(仕様書内)				12
		C-2	階層主項に対しての記載矛盾(仕様書内)				68,40,51
C-3		同一主項に対しての記載矛盾(仕様書内)				66,23	
C-4		同一主項に対しての記載矛盾(仕様書内)				31,45	
D 誤記	D-1	プログラム実装に 対する記載ミス				3,4,44	
	D-2	利用手順書に 対する記載ミス				68,21,29,16,17	
E 曖昧	E-1	用語の定義				36	
	E-2	時間の定義				36,53	
	E-3	主辞の定義				64	

図 3 表計算ソフト形式に変換した「観点 60」

表計算ソフト形式では「観点 60」は二次元表となっており、縦の列に観点、横の列に機能が記載してある。

例えば、ユーザビリティの観点から入力機能を確認すると入力時の視認性、理解容易性といった仕様書の指摘予備軍が潜在している箇所を導き出すことができる。

実際に「観点 60」を使用することで、仕様書レビューの課題である「属人化からの脱却」、つまり「業務経験が浅いメンバと経験豊富なメンバとの差を抑え、同等の指摘数をあげること」を解消することができるのだろうか。

そこでその効果を検証すべく、実証実験を行った。

#### 4. 実証実験

実証実験の方法としては、以下の 3 グループに被験者を分け、かつ、「観点 60」を使用した場合と「観点 60」を使用しなかった場合でどれだけ指摘件数に違いが発生するかを検証した。

- テスト設計の経験が 2 年未満 ……①(A, D)
- テスト設計の経験が 3 年～6 年未満 ……②(B, E)
- テスト設計の経験が 7 年以上 ……③(C, F)

上記①～③に該当するメンバを 2 名ずつ選出し、さらにその 2 名を「観点 60」を使用した場合(A, B, C)」と「観点 60」を使用しなかった場合(D, E, F)」に分け、計 6 名を対象に実証実験を実施した。

また、被験者が実証実験に費やした時間により、指摘件数に差異が発生することを防ぐため、制限時間を設定した。今回、実証実験を行うにあたっての制限時間は 90 分間とした。

なお、今回の実証実験は『ASTER 自動販売機ユーースケース仕様書「2. 商品選択ユーースケース」』[1]と『ASTER 自動販売機ハードウェア構成および販売者用機能仕様』[2]を対象に問題点の指摘を行った。

#### 5. 実験結果

実証実験の結果を表 1 に示す。

表 1 仕様書の問題点を指摘した件数の比較

	①	②	③	平均
「観点 60」使用	11(A)	15(B)	16(C)	14.0
「観点 60」未使用	6(D)	7(E)	11(F)	8.0
指摘件数比率	83%up	114%up	45%up	75%up

結果として、どのグループも「観点 60」を使用した方が多くの指摘をあげられたことがわかる。

注目すべきは、「観点 60」を未使用の(F)と「観点 60」を使用した(A)の指摘件数が同数であるということである。

また、①、②グループは、「観点 60」を使用した方が、「観点 60」未使用より約 2 倍の指摘数を発見しており、③グループと比較して、より効果が発揮されていることがわかる。

②グループに至っては、①グループよりも発見比率が高く、(B)が(F)とほぼ同数の指摘数を発見している。このことから最も効果を発揮できるのは、②のテスト設計の経験が 3 年～6 年未満のグループということも注目すべき点である。

以上のことから、「観点 60」を使用した場合に経験の浅いメンバでも経験豊富なメンバと同程度の指摘をあげられることが実証できたとと言える。

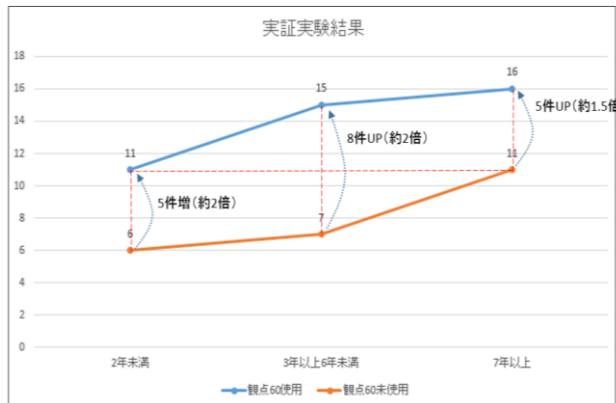


図 4 実証実験の結果

## 6. 考察

### 6-1. 有効性

実験結果より、「観点 60」未使用時と比較して、「観点 60」を使用すると全体で約 1.75 倍多くの指摘をあげることができた。

これによりテスト設計経験の層に依存することなく仕様書の指摘数を多く報告することが可能となることがわかり、仕様書レビューにおいて「観点 60」の有効性が高いことが実証できた。

中でも知識・技術やノウハウが十分とはいえないが、ある程度の経験はあると考えられるテスト設計経験 3 年～6 年未満の層が最も高いパフォーマンスを発揮することがわかる。

このことから、いくらツールに有効性があったとしても個人の技術なくしてはツールの効果を十分に引き出すことはできないことが考えられる。

例として、この結果を現実的なプロジェクト事例に置き換えて考える。

仕様書レビューを 10 人規模のプロジェクトで実施すると仮定し、メンバの内訳を(A)が 5 人、(B)が 3 人、(C)が 2 人と設定する。

図 4 実証実験の結果の結果から、通常では 10 件の指摘件数に対して、「観点 60」を使用することで、それよりも 1.75 倍多い、約 17 件の指摘をあげられることが予測される。

### 6-2. テストへの応用

「観点 60」の有効性を実証していく中で、実際の指摘結果を抽象化し、それを分類したことにより属人化を防ぐ特性を持つ「観点 60」は、仕様書の指摘にとどまらず、それ以外に活用できると考えられる。

「観点 60」の各観点でどのような指摘がされているか内容を精査すれば、どの観点からどのような指摘を

多く検出することができるかを予想することができる。

また、各観点に重み付けを行えば観点ごとによる重要度を計測でき、指摘の傾向分析を行うこともできる。

さらに応用したパターンを述べる。例として、テスト実行フェーズで実施する探索的テストで検出された不具合に「観点 60」を用いてみた場合を考える。

指摘された不具合から観点を抜き出し、「観点 60」の作成と同様に収集・解析しデータベース化すれば、テスト実行者の能力に依存しがちである探索的テストにおいて、経験値の差に依存する割合を抑え、より多くの不具合の発見ができるツールになりえる可能性がある。

### 6-3. 今後の課題

今回、本稿を作成する過程で着手できなかった重要な課題がある。今後「観点 60」の精度を上げていくために優先して取り組むべきタスクとして、以下の 3 点を述べる。

#### 6-3-1. 指摘内容の妥当性

今回の実証実験では、問題点の指摘件数のみを計測した。指摘件数増加という効果を得られることはわかったが、指摘内容の妥当性、つまり指摘の“質”については考慮されていない。

有効性を高めるためにも指摘した内容が、ソフトウェア開発の効率化にどの程度影響を与えているのか検証する必要がある。そのためにも指摘結果を 1 件ずつ分析し、“質”を確かめる必要がある。

今後はまずその“質”の計測方法から検討し、指摘の比較方法を決め分析する。

#### 6-3-2. 具体的効果の実証

今回の実証実験では、「観点 60」使用時の指摘件数を計測することはできたが、その指摘が実際にどの程度価値があり、役立つのかまでは考慮されていない。そこで「観点 60」を使用したことによる「具体的な効果の実証」を行う必要がある。

実証実験のように制限時間を設けず、十分に人数と時間を掛けて今回使用した仕様書に対して指摘されるべき指摘一覧といった模範解答を作りそれと指摘結果を照らし合わせる。

それにより、「どこまで短時間で模範解答に近づけたか」、または、「経験の差に依存せず質の高い指摘をあげることができたのか」を明確にすることで「観点 60」の具体的な効果を実証したといえる。

#### 6-3-3. 最適な使用方法の追求

今回の実証実験では、観点 60 を使用するにあたり発想を狭めないために被験者に細かい使用方法是指示していない。ただ、仕様書レビューの際に「観点 60」を用いて実施することだけを伝え、実験を行った。

実際に検証実験に参加したメンバに「観点 60」の使用法や使用したタイミングをヒヤリングしたところメンバごとに違いが見られた。

例えば、最初は「観点 60」を使わず、仕様書を読み込んだ後に最終チェックとして「観点 60」で自分のあげた指摘を確認するケースや、1 行読んだら「観点 60」を見る、を繰り返して常に意識して使用するケースなどメンバごとに異なった使い方がされていた。

より多くの指摘をあげることのできる最適な使用法を確立しなければ、結果として「観点 60」も属人化したツールとなってしまう。

今後は最適な使用法を確立することも考慮に入れ、実証実験を行う必要がある。

## 7.まとめ

最後に、本稿で明らかになったことと、限界を明記するとともに今後の研究の方向性を述べる。

本稿では、仕様書レビューを実施する際に、メンバの知識・技術に関わらず、誰もが同程度の成果をあげるための観点階層図として「観点 60」を提案した。

また「観点 60」の有効性を検証するための、実証実験、および実験結果を示した。

この結果から、「観点 60」は仕様書の指摘を実施する際に、メンバに依存することなく、指摘をあげることができる観点階層図として有効であることが明らかになった。

また、本稿の限界として今回の実証実験は 6 人という少人数を対象とした実証実験しかできていないことがあげられる。有効性、正当性を高めるためにも今後は今回の倍以上の人数を対象に実証実験を実施する。

本稿で実施している実証実験の分析では、指摘数のみに主眼を置いており、指摘の内容は考慮されていない。そのため、具体的な効果の証明には至っていない。

また指摘内容の妥当性を検証するために、模範解答を作成する必要性や「観点 60」の最適な使用法を確立する必要がある。

以上のことから、今後重要視する点は以下の 3 点があげられる。

- 「観点 60」の指摘内容の“質”を計測できる仕組みを作成する
- 最適な使用法を実証できる実験方法を検討する
- 十分に時間を掛け作成した模範解答を用意する

今後の方向性としては、これらを考慮して再度実証実験を実施することである。

## 参考文献

[1] ASTER、自動販売機ユースケース仕様書 P.6-9「2.商品選択ユースケース」、2013

[2] ASTER、自動販売機ハードウェア構成および販売者用機能仕様、2013