Two-level Checklists and Perspectives: Software Reading Techniques for Pattern Writer's Workshop

TIAN XIA, Waseda University
HIRONORI WASHIZAKI, Waseda University
YOSHIAKI FUKAZAWA, Waseda University
JOSEPH YODER, The Refactory, Inc.
REBECCA WIRFS-BROCK, Wirfs-Brock Associates, Inc.

A Pattern Writer's Workshop (WW) is a peer-review method to improve pattern or pattern language papers. However, several problems may arise, leading to "ad hoc" review meetings where patterns are not systematically reviewed and authors do not receive useful feedback. Previously, we considered Checklist-based Reading (CBR) and Perspective-based Reading (PBR), which are techniques to improve ad hoc reviews, but some deficiencies were noted. In this paper, we propose an approach to improve WWs by providing a two-level checklist and refining perspectives for reviewing patterns. An experiment shows the strengths and weaknesses of this new approach.

Categories and Subject Descriptors: **D.2.11 [Software Architectures]: Patterns**

General Terms: Management, Documentation, Performance, Design, Experimentation, Human Factors

Additional Key Words and Phrases: Software pattern, Pattern language, Writer's workshop, Review, Checklist-based reading, Perspective-based reading

ACM Reference Format:

Pineo, D. and Ware, C. 2010. Neural Modeling of Flow Rendering Effectiveness. ACM Trans. Appl. Percept. 2, 3, Article 1 (May 2010), 10 pages.

1. INTRODUCTION

Pattern Writer's Workshops (WWs) are designed to review and improve pattern or pattern language papers via feedback from peers. Although WWs are well accepted in the pattern community and xPLoP conferences, several problems may arise due to how they are conducted, leading to an "ad hoc" review meeting where the patterns are not adequately reviewed and the authors do not receive useful feedback. Software reading techniques such as Checklist-Based Reading (CBR) and Perspective-Based Reading (PBR) have been developed as improvements over ad hoc reviews of software and software documentation. Because software patterns and pattern languages are software artifacts, applying existing software reading techniques to review software pattern or pattern language papers may improve the outcome of WWs.

Previously we proposed an approach to introduce CBR and PBR to WWs. Our experiment, which examined their contributions, identified several issues. First, our CBR checklist was unsuited for all types of pattern papers; some items were not applicable and other important items were missing. Second, some participants expressed difficulty understanding certain items because our list was a mixture of abstract topics and concrete items. Third, we were unable to confirm an obvious contribution of PBR during the experiment. Fourth, a more precise process was highly desired by all participants.

Herein we propose an approach to provide a two-level CBR checklist and to revise the perspectives in PBR. Additionally, we recommend a process showing how to utilize these reading techniques in WWs that target patterns and pattern languages. To validate the effectiveness of our approach, an experiment was conducted to address the following research questions:

RQ1. Does the two-level checklist increase the number of general comments compared to a single checklist or a traditional WW?

RQ2. Does the PBR provide more specific comments than a traditional WW?

RQ3. Is our process more efficient than a traditional WW or a single checklist?

The rest of the paper is structured as follows: Section 2 introduces our previous research and the identified problems in detail. Section 3 describes the proposed CBR and PBR and shows the process to utilize them.

Section 4 presents an experiment to validate our approach and answer the research questions. Finally, Section 5 draws conclusions and provides research directions.

2. BACKGROUND AND PROBLEM STATEMENT

2.1 Previous research

Although WWs are accepted widely, there can be several problems with the outcome, such as too few comments, superficial comments, or missing important concerns (Fig. 1). In our previous research, which examined the causes of these problems, we proposed an approach to introduce CBR and PBR into WWs targeting patterns and pattern languages. The main ideas and contributions can be summarized as follows:

- CBR: To review software, CBR is a reading technique where reviewers use a list of statements or questions when checking a document. To apply CBR to WWs, we surveyed the existing literature on software pattern writing, shepherding, and the criteria for well-written software patterns. Then we composed a checklist of questions regarding the properties that a pattern should possess. To cover all aspects of a pattern, the checklist contained 27 questions in 10 categories. For example, questions included “Does the pattern contain a pattern name, context, problem, system of forces, and solution?” in the “Structure” category and “Does the problem and solution match and fit together?” in the “Problem and Solution” category.

- PBR: PBR asks reviewers to provide comments based on an assigned perspective. Previously, we applied PBR to WWs by combining PBR and CBR. That is, we made checklists from several perspectives. For example, the “End User” checklist asked “Does the pattern help satisfy the needs and requirements in the resulting software” and “Does the pattern contribute to the ease of use of the resulting software?” These questions were intended to deepen the reviewer’s consideration of the pattern content based on a specific perspective.

Moreover, to validate the effectiveness of our approach, the previous paper included an experiment. The results confirm that CBR produces more comments than a traditional WW with regard to the description and pattern contents, and that PBR might contribute to more concrete content.

2.2 Problems and possible solutions

Although the previous research showed that our approach had some positive contributions, several problems were also revealed, including:

- P1. Difficulty understanding the checklist for CBR: We strived to create an easy-to-use general checklist based on several categories, but many comments indicated the checklist was difficult to use.
  Cause*:* Some items were described too abstractly while other items were simply yes or no. This mixed format made the list difficult to understand. Additionally, some unnecessary items also caused a problem (related to P3).

- P2. No obvious contribution of PBR: We were unable to validate the contribution of PBR compared to a traditional WW.
  Cause: There were several possible reasons. First, the number of perspectives was limited, restricting the number of comments. Second, some perspectives were hard to use for certain patterns (related to P3). Third, the primary benefit of PBR might not be to find more comments, but to find more specific comments.

- P3. Unsuitable checklists in both CBR and PBR: Some items in the CBR checklist and some perspectives in PBR were unsuited for the target pattern, leading to other issues.
  Cause: This problem was due inflexibility. Because numerous patterns exist, the fixed checklist was not universal.

- P4. Ambiguous Processes in both CBR and PBR: The moderator and participants were confused by receiving the checklists without instructions. They were unsure on how to apply them to a WW.

Cause: A precise explanation of the whole process was lacking.

In this paper, the following solutions are proposed.

- S1. Two-level Checklist-based Reading
- S2. Revised perspectives
- S3. Process to utilize CBR and PBR in WWs

Figure 1 shows the relationship between the problems and the solutions. Below they are described in more detail.

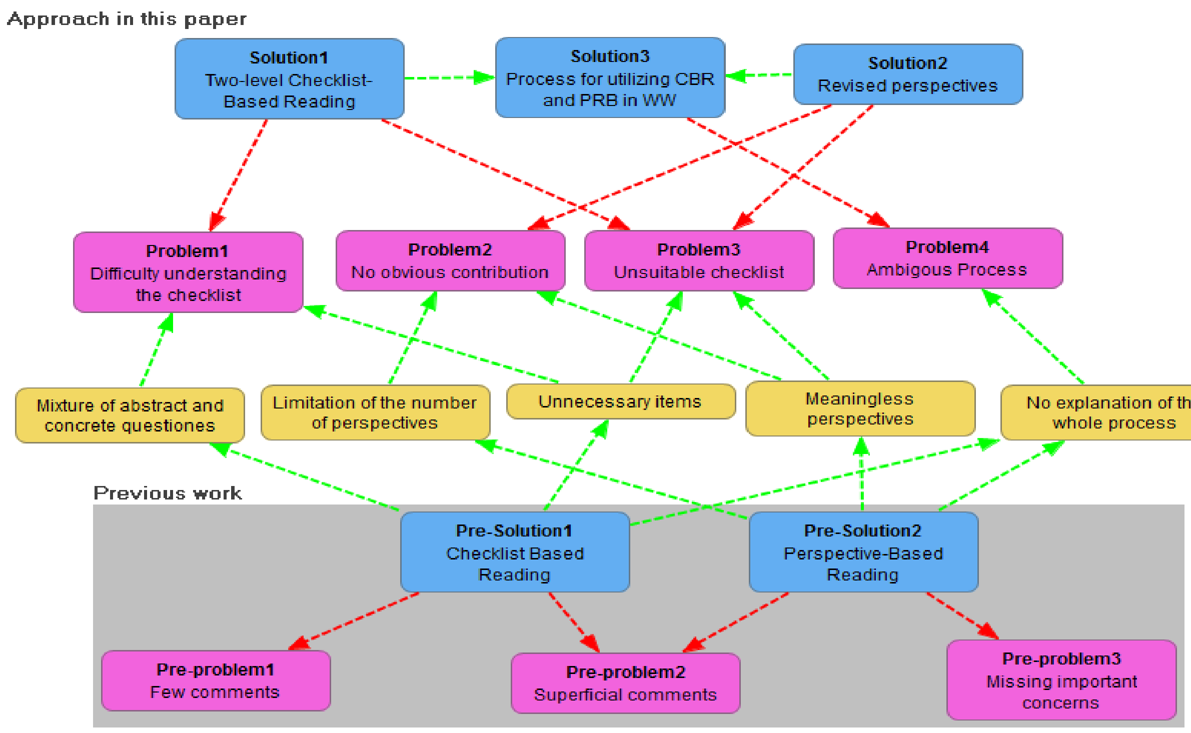
Approach in this paper
Solution1 Two-level Checklist-Based Reading
Solution3 Process for utilizing CBR and PRB in WW
Solution2 Revised perspectives
Problem1 Difficulty understanding the checklist
Problem2 No obvious contribution
Problem3 Unsuitable checklist
Problem4 Ambigous Process
Mixture of abstract and concrete questiones
Limitation of the number of perspectives
Unnecessary items
Meaningless perspectives
No explanation of the whole process
Previous work
Pre-Solution1 Checklist Based Reading
Pre-Solution2 Perspective-Based Reading
Pre-problem1 Few comments
Pre-problem2 Superficial comments
Pre-problem3 Missing important concerns

Figure 1 Relationships between the problems, causes, and proposed solutions

3. READING TECHNIQUES FOR WW

Here we propose an approach to introduce two representative reading techniques, CBR and PBR, into traditional WWs as well as process on how to utilize them.

3.1 Two-level Checklist-based Reading

Two types of checklists can be used in WWs: general and specific checklists. A general checklist is more abstract, focusing on the structure and description of the pattern, while a concrete checklist emphasizes details of the pattern content. In this paper, we provide a two-level checklist that considers both types.

3.1.1 Method

A “two-level” checklist includes both abstract and concrete levels for a comprehensive overall review. One level probes abstract ideas and considerations, while the other shows concrete items and issues to be checked. To create such a checklist, we surveyed the existing literature. Items are classified into the following ten categories.

- Being generative: According to the original concept of patterns and pattern languages, patterns are generative. Although generativity is a desired characteristic of any software pattern or software pattern language, we have found that reviewers have difficulty answering such a conceptual question easily. Hence, we generated several items related to how patterns can be generative.

- Domain and scope: A pattern, as a part of a larger pattern language, should focus on a specific scope within a domain.
- Structure: A pattern should contain several mandatory elements such as name, problem, and solution. A pattern should also have optional elements, including code examples, resulting context, and related patterns.
- Problem and solution: A problem and its corresponding solution are the heart of a pattern. They should work well individually and together.
- Forces: Forces should be addressed comprehensively and visibly support the problem.
- Name and reference: A pattern should have the appropriate name and refer to other patterns explicitly.
- Known uses and validation: A pattern should be validated by use.
- Acknowledgement: Authors of a pattern should recognize people who contribute to their pattern.
- Terminology and notation: A pattern should be comprehensive and use common terminology and figure notations.
- Pattern language: A pattern language, as a system of related patterns, should have a precise summary and ideally a common running example for all patterns within it.

3.1.2 Two-level checklist

Table 1 shows the abstract level of our general checklist. Because the items are described in an abstract manner, this table is suitable for reviewing most types of patterns or pattern languages. This abstract checklist may be a solution to our "unsuitable checklist" problem (P3).

Table 1 Draft of the abstract level for a general checklist

ID	Category	Item to be checked
A1	Being Generative	Is the pattern both a thing and a process?
A2	Being Generative	Does the pattern have an implied artifact?
A3	Being Generative	Does the pattern realize many levels of abstraction?
A4	Domain and scope	Is the pattern grounded in a specific domain and as part of a language?
A5	Domain and scope	Is the pattern target clear?
A6	Structure	Does the pattern contain all necessary information?
A7	Structure	Would adding additional information be meaningful or helpful?
A8	Structure	Does the pattern help the reader catch the essence quickly?
A9	Problem and solution	Do the problem and solution work well separately?
A10	Problem and solution	Do the problem and solution match and work well together?
A11	Forces	Does the pattern address all forces comprehensively?
A12	Forces	Do the forces clearly lead to the choice of solution to the problem?
A13	Name and reference	Is the pattern name meaningful and easily remembered?
A14	Name and reference	Does the pattern refer to other external patterns in understandable ways?
A15	Known uses and validation	Is the pattern validated sufficiently?
A16	Acknowledgement	Do the authors acknowledge those who supported their pattern writing?
A17	Terminology and notation	Does the pattern use terminology and notations in a comprehensive way?
A18	Pattern language	Is each pattern in the language defined?
A19	Pattern language	Are the relationships between patterns in the pattern language clear?
A20	Pattern language	Does the pattern language help the reader put this language into practice?

This checklist may not solve the problem, "difficulty understanding the checklist" (P1). Abstract items may make the checklist more difficult to understand and confuse participants who are unfamiliar with WWs targeting patterns.

Table 2 shows the concrete level for the general checklist. Each category contains more items than the abstract level. To a certain extent, the concrete checklist could be considered a detailed example of the abstract checklist. Most of the items in this checklist are readable and easy to understand. Even WW participants with little experience reviewing patterns should be able to provide valuable comments following this list. Thus, the list in Table 2 is a solution to "difficulty understanding the checklist" (P1).

ID	Category	Item to be checked
C1	Being Generative	Does the pattern introduce a good system as well as the process to build it?
C2	Being Generative	Does the pattern show different levels (e.g., story, paragraph, sentence, and word)?
C3	Being Generative	Are the design levels linked?
C4	Being Generative	Does the pattern leave an inevitable mark on the structure of its application result?
C5	Domain and scope	Does the pattern clarify the domain it serves?
C6	Domain and scope	Is the pattern connected with other patterns?
C7	Domain and scope	Is the pattern part of a pattern language?
C8	Domain and scope	Is a target audience of the pattern clear?
C9	Domain and scope	Does the pattern have the proper scope for its application target?
C10	Structure	Does the pattern contain a pattern name, context, problem, system of forces, and solution?
C11	Structure	Are problem, context, and solution identified clearly if a reader only skims the paper?
C12	Structure	Is the pattern readable upon skimming?
C13	Structure	Does the pattern contain additional information such as the resulting context, running examples, and related patterns?
C14	Structure	Do the additional sections provide sufficient information about the need for the pattern?
C15	Problem and solution	Does the problem capture system hot spots?
C16	Problem and solution	Does the solution focus on an area rather than a one-time problem?
C17	Problem and solution	Do the problem and solution match?
C18	Problem and solution	Do the problem and solution provide the core idea of the pattern?
C19	Forces	Does the pattern address both functional and nonfunctional forces?
C20	Forces	Do the forces explain what makes the problem difficult?
C21	Forces	Are the forces highly visible regardless of the pattern form used?
C22	Name and reference	Is the pattern named by its solution or a meaningful metaphor?
C23	Name and reference	Is there a brief explanation of a related pattern introduced in the paper?
C24	Name and reference	Is the relationship between the pattern and related pattern(s) presented clearly?
C25	Known uses and validation	Is the pattern validated by use, preferably at least three times?
C26	Known uses and validation	Are the known uses or stories convincing?
C27	Acknowledgement	Do the authors acknowledge others for their shepherd and workshop applicants?
C28	Terminology and notation	Does the pattern use terminology and notations that the audience will understand?
C29	Terminology and notation	Is a glossary of unfamiliar terms provided, if necessary?
C30	Pattern language	Is the summary of each pattern provided in terms of its problem and solution?
C31	Pattern language	Is the pattern language summarized in the introduction?
C32	Pattern language	Does the pattern language show a process to use the provided patterns?
C33	Pattern language	Is the common problem highlighted if several patterns solve same problem?
C34	Pattern language	Is the same running example used through the entire language?

Table 2 Draft of the concrete level for a general checklist

More concrete questions lead to an inflexible review. This concrete checklist might not solve the "unsuitable checklist" (P3), and may even make the problem worse. Additionally, important concerns may be missed if the concrete checklist does not cover important aspects of a pattern.

To summarize, both checklists may solve one problem while making another problem worse. Because this is highly undesirable, a combination of both checklists is considered.

3.1.3 Usage of the checklist in WWs

We suggest using the two-level checklist as follows. While preparing for a WW, the authors of the pattern prepare a tailored checklist based on the checklists in Tables 1 and 2. If a concrete checklist for each category is appropriate, then the concrete questions should be used. However, if the concrete checklist only partially fits their pattern or pattern language, unsuitable items should be modified. However, if the concrete items are inapplicable, the abstract checklist should be considered. By combining questions for each category from both checklists, the authors can generate a customized checklist to review their pattern or pattern language. Table 3 shows an example of a customized checklist.

ID	Category	Item to be checked
C1	Being Generative	Does the pattern introduce a good system as well as the process to build it?
C2	Being Generative	Does the pattern show different levels (e.g., story, paragraph, sentence, and word)?
C3	Being Generative	Are the design levels linked?
C4	Being Generative	Does the pattern leave an inevitable mark on the structure of its application result?
C5	Domain and scope	Does the pattern clarify the domain it serves?
C6	Domain and scope	Is the pattern connected with other patterns?
C7	Domain and scope	Is the pattern part of a pattern language?
C8	Domain and scope	Is a target audience of the pattern clear?
C9	Domain and scope	Does the pattern have the proper scope for its application target?
C10	Structure	Does the pattern contain a pattern name, context, problem, system of forces, and solution?
C11	Structure	Are problem, context, and solution identified clearly if a reader only skims the paper?
C12	Structure	Is the pattern readable upon skimming?
A7	Structure	Would adding additional information be meaningful or helpful?
A8	Structure	Does the pattern help the reader catch the essence of it quickly?
C15	Problem and solution	Does the problem capture system hot spots?
C16	Problem and solution	Does the solution focus on an area rather than a one-time problem?
C17	Problem and solution	Do the problem and solution match to each other?
C18	Problem and solution	Do the problem and solution provide the core idea of the pattern?
C19	Forces	Does the pattern address both functional and nonfunctional forces?
C20	Forces	Do the forces explain what makes the problem difficult?
C21	Forces	Are the forces highly visible regardless of the pattern form used?
C22	Name and reference	Is the pattern named by its solution or a meaningful metaphor?
C23	Name and reference	Is there a brief explanation of a related pattern introduced in the paper?
C24	Name and reference	Is the relationship between the pattern and related pattern(s) presented clearly?
C25	Known uses and validation	Is the pattern validated by use, preferably at least three times?
C26	Known uses and validation	Are the known uses or stories convincing?
A16	Acknowledgement	Do the authors acknowledge those who supported their pattern writing?
A17	Terminology and notation	Does the pattern use terminology and notations in a comprehensive way?

Table 3. Example of a customized checklist for CBR

One benefit of this approach is that a customizable checklist mitigates P3 while simultaneously providing a solution to P1 as the tailored checklist is mostly based on concrete items.

3.2 Perspective-based Reading

3.2.1 Method

Existing perspectives for reading software materials are also applicable to WWs for software patterns and pattern languages. Such perspectives include quality characteristics, use cases, usage scenarios, and stakeholders. Possible uses of these perspectives were discussed in our previous research. Although our previous work incorporated perspectives of the stakeholders of the resulting software, this paper also provides perspectives of stakeholders of the pattern and pattern language. Tables 4 and 5 show checklists for possible stakeholders.

Stakeholder	ID	Considerations
End user	E1	Does the pattern satisfy needs and requirements in the resulting software?
	E2	Does the pattern help reveal possible users and behaviors of the resulting software?
	E3	Does the pattern help improve the ease of use of the resulting software?
Designer	D1	Does the pattern provide sufficient and consistent information for design?
	D2	Does the pattern contribute to an adequate design solution for the resulting software?
	D3	Does the pattern allow for future extensions and maintenance of the resulting software?
Programmer	P1	Does the pattern provide sufficient information to show it solves the problem?
	P2	Is the pattern adaptable to most software implementations regardless of the type of software?
	P3	Does the pattern provide a solution without unnecessarily increasing software complexity?

Tester	T1	Is the mechanism of the pattern solution reliable?
	T2	Can the pattern solution be easily tested?
	T3	Does the pattern contribute to testing ease of the resulting software?
	T4	Does the pattern provide sufficient information to test the resulting software?
	T5	Does the pattern contribute to robustness of the resulting software for any input?
Quality engineer	Q1	Does the pattern show how it contributes to the quality of the solution software?
	Q2	Does the pattern avoid introducing additional risks or decrease the quality of the solution software?

Table 4.Draft checklist for stakeholders of the resulting software

Stakeholder	ID	Considerations
Author of related pattern	AR1	Is the explanation of the related pattern correct and clear?
	AR2	Does the pattern refer to the core idea of related patterns?
Author of another pattern paper	AP1	Is the core idea of the pattern clear enough that it could be referred to easily?
	AP2	Does the pattern show how it could contribute to related patterns?

Table 5. Draft checklist for other pattern stakeholders

Because a pattern should be able to integrate into its surrounding environment, the perspective of authors of other patterns is important. Thus, Table 5 shows a checklist for two additional types of stakeholders:

- Author of a related pattern: A related pattern is defined as any pattern related to the target pattern. The author of a related pattern may care about whether their pattern is referred to correctly and meaningfully.
- Author of another pattern paper: Because authors of other patterns may refer to the target pattern, they should be able to easily identify the core idea of the target pattern in order to determine whether the target pattern is useful as a reference or part of their solution.

Unlike CBR, PBR focuses more on pattern content than on form or description. Because it is difficult to provide concrete items to check pattern content, we instead provide several considerations for each stakeholder. This checklist reminds participants of their character and points out possible considerations.

By increasing the number of stakeholders, the problem, “No obvious contribution” (P2) might be mitigated, as more specific views are represented. However, each pattern or pattern language has its own target audience. Stakeholders for the resulting software are highly dependent on the pattern itself. Thus, reviewing the perspectives of all potential stakeholders might not be meaningful. Moreover, due to time limitations, unsuitable perspectives may waste resources, reducing comments from more relevant perspectives.

Therefore, we strongly recommend that the authors of a pattern or pattern language select only the essential perspectives from the Table 4 (mitigation for P3), and depending on the pattern, the authors may add specific perspectives.

3.3 Process to utilize reading techniques in WWs

We propose the following implementation process to improve WWs:

0) Prior to the WW, the authors of the pattern prepare a customized checklist and perspectives based on 3.1.3 and 3.2. The moderator confirms the checklists. Then the participants carefully read the paper and prepare comments.
1) The moderator assigns both the customized checklist and specific perspectives to the participants.
2) The author of the pattern under review reads one or two important paragraphs in the pattern paper.
3) One of the participants summarizes the paper.
4) Participants identify the strengths of the pattern by CBR. The moderator leads a discussion on general items according to the customized checklist.
5) Participants identify the strengths of the pattern by PBR. The participants comments based on their perspective.

6) The same process as (4)–(5) is applied to identity the pattern's weaknesses and to suggest possible improvements.
7) A free form discussion ensues to capture comments that not covered in the previous discussions.
8) The author thanks all the participants and asks questions to clarify the participants' statements.
9) The participants clap to thank the author for writing the paper.
10) The participants submit the draft paper with comments to the author, if necessary.

Because this process follows a traditional WW, the authors should observe without commenting during the main discussion. The benefit of this process is that both the pattern structure and its contents are more systematically evaluated. CBR allows participants to have a systematic discussion on the structure and description, while subsequently employing PBR after most superficial issues have been discussed allows participants to focus on the pattern content.

4. EXPERIMENT AND DISCUSSION

To validate the effectiveness and efficiency of our approach, and answer RQ1-RQ3, we conducted a local experiment at Waseda University.

4.1 Experimental Setup

To compare to a traditional WW, we choose a pattern called "Enterprise Service Bus (ESB)", which was reviewed by a traditional WW in PLoP2011 and used in our previous work. Thus, we believe the results will clearly show the effectiveness of our approach. The experiment involved with four participants, including three master students and the first author of this paper. All students had some knowledge about patterns, but some were not familiar with the WW format. Thus, we gave a brief explanation before starting the mini WW as well as undertook the following preparation steps prior to the experiment:

- CBR: A customized checklist was prepared by combining the two-level checklists (Table 3).
- PBR: Three perspectives (designer, tester, and quality engineer) were chosen.

4.2 Experiment Results and Discussion

Figure 2 shows the number of comments received in each review, including the original WW in PLoP2011, our previous work, and the results of this experiment. Detail comments could be found in table 7 and 8 (Appendix). To answer RQs 1 and 2, the comments were categorized based on whether they focused on a general topic (structure and form) or pattern details and specific contents.

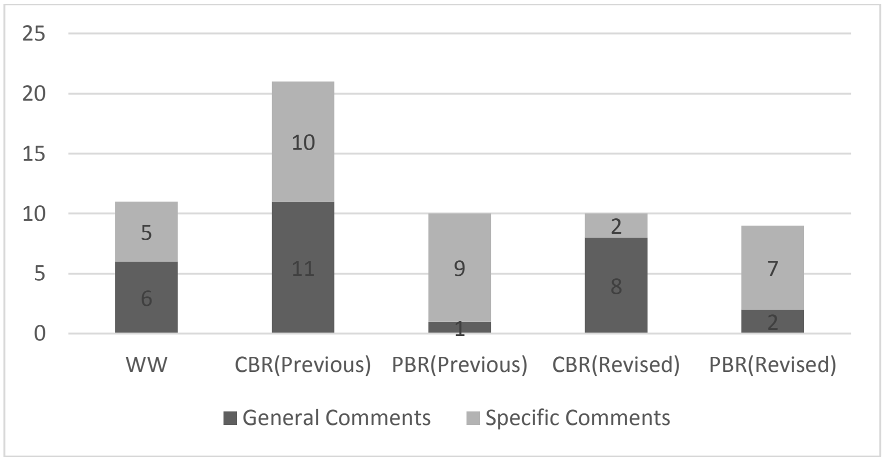

Figure 2.Number of comments received for each review process

To illustrate the efficiency of our process, Table 6 shows the reviewing time costs and the calculated the efficiency, which is determined by dividing total number of comments by the time.

Process	Total Comments	Time (min)	Efficiency
WW	11	40	0.275
CBR (previous)	21	45	0.467
PBR (previous)	10	45	0.222
Our process (CBR&PBR)	19	40	0.475

Table 6.Time cost and efficiency of each review process

RQ1. Does the two-level checklist increase the number of general comments compared to a single checklist or a traditional WW?

According to Fig. 2, eight of the ten comments were general ones using CBR, which is more than from the traditional WW. Although the CBR comments provided specific feedback such as the "target audience is unclear," most comments were general comments, demonstrating that the two-level checklist helps participants provide general comments about the form and structure of the pattern. In contrast, there were fewer general comments than in our previous experiment. This discrepancy is likely due to the way comments were recorded. Our most recent experiment did not count simple comments such as, "Yes, it has a good name" as these may not be valuable to the author. However, several of these simple comments were counted in our previous work. These results show that a two-level checklist is more effective than a traditional WW but not necessarily an improvement compared to a single checklist.

RQ2. Does the PBR provide more specific comments than a traditional WW?

Compared with the original WW, PBR helps participants focus on detailed content and give more specific comments. Participants were able to review the pattern details based on the assigned perspective and discuss the pattern with a deeper understanding. However, we noticed that the number of comments did not increase; actually, there were fewer comments compared to our previous approach even though the problem of unsuitable perspectives is resolved. In addition, we were unable to verify the utility of the perspectives for the stakeholder of a pattern (i.e., Table 5) because our participants are neither authors of other patterns nor related patterns. Although our PBR technique provides more specific comments, some problems, which require further study, may still exist.

RQ3. Is our process more efficient than a traditional WW or a single checklist?

According to Table 6, our experiment produced 19 comments in 40 minutes. The proposed PBR is nearly twice as efficient as a traditional WW and our previous PBR. Although the two-level checklist has an efficiency similar to the previous single-level CBR, some simple comments were not recorded during the latest experiment. Thus, the two-level checklist more efficiently generates comments than a traditional WW and our previous research experiments.

4.3 Experiment Summary

The results did not fully meet our expectations. According to RQ1 and RQ2, we failed to confirm any improvement over our previous work because fewer comments than expected were generated. On the other hand, several contributions of a two-level checklist were confirmed. First, participants easily applied our process after only a brief explanation, resulting in the best efficiency (RQ3) and resolving the problem of an "ambiguous process" (P4). Second, the time duration of our process is almost same as traditional WW, suggesting that our approach could easily replace traditional reviews. Third, the customized checklist and perspectives are beneficial. Participants understood the checklist content, even if it was their first WW experience. Fourth, all the participants confirmed that their assigned perspectives for PBR were meaningful for reviewing the pattern and that the checklist assisted in providing meaningful comments.

In conclusion, our approach realizes a process to successfully review pattern papers in a WW and generates more comments in the same amount of time as a traditional WW. Additionally, our approach is easy to understand and resolves several problems noted in our previous research. However, if the authors prefer to receive comments as many as possible, it is unclear whether the two-checklist approach is more beneficial than a single-checklist approach.

5. CONCLUSION

Herein we propose an approach, which creates a two-level checklist for Checklist-Based Reading (CBR) and extends Perspective-Based Reading (PBR), to solve several problems noted in our previous research that introduces reading techniques into WWs. We also show how to use each review technique and describe the process for an improved WW (including its preparation) using our checklists. A small experiment confirmed the benefits of a two-level checklist in resolving some previous issues.

In the future, we plan to continue studying this topic. Specifically, we plan verify the usefulness of different stakeholders, make the checklist more customizable according to pattern authors' requirements, conduct additional experiments to verify and fine-tune our approach, and identify the merits and drawbacks. We are planning a focus group at EuroPLoP 2015 and conducting an experiment aimed at improving our review techniques.

6. ACKNOWLEDGMENTS (TBD)

Our thanks go to shepherd Jason Yip for his careful review that improved this paper. We also thank to the participants from Washizaki's Lab at Waseda University (Frederik Nakstad, Harlin Ismail Rizky, Rizki Amelia) who participated in the case study.

APPENDIX

In order to show the difference between traditional WW and our experiment, we conclude the comments in each case in the following table.

Comments
This article explain the pattern clear by using several figure
Article has a good structure (people can find traditional parts like example, force, problem and solution in the article)
ESB can be accepted to some element pattern learner
It matches the force and the problem
Force is given like some one line requirement, which divide a difficult problem to several simple part
No detail way about how to use ESB
This article is introducing rational but not implementation
A service can be connected to the system as every company use their own server for ESB
Multiple ESB server can be considered
There are no exact example for ESB
Cost of ESB should also be discussed (manager want to know it)

Table 7. Comments received in traditional WW (PLoP2011)

	Category	Comments
CBR	Domain and Scope	Figure 4 shows the relationship with relation pattern
	Structure	Many figure are used for explanation
	Structure	Dynamic figure is good for showing extra information
	Forces	Force is shown clear and readable
	Know uses and reference	This pattern has strong known usage by giving example of some big companies
	Domain and scope	It's better to declare its audience clear in the paper
	Structure	No implement example is provided
	Problem and solution	Problem does not catch enough. More information should be provided.
	Forces	Force could be arranged by functional and nonfunctional
	Forces	The relationship of solution and force are not shown clear

	Perspective	Comments
PBR	Designer	This pattern is shown clear and can be quickly understand for designer
	Designer	There are several important concerns existing in forces.
	Tester	System is not so complicate for test
	Tester	This pattern is more like a module solution
	Tester	Hardness of testing the ESB should be considered
	Quality Engineer	Its Force show how it contribute to increase quality well.
	Quality Engineer	Liability is good on showing some potential risk
	Quality Engineer	This pattern might discuss more about its effectiveness on quality in the paper.
	Quality Engineer	This pattern didn't show the motivation of dealing with the liability

Table 8. Comments received in our case study

REFERENCES

Hironori Washizaki, Tian Xia, and Yoshiaki Fukuzawa, "Introducing Software Reading Techniques into Pattern Writer's Workshop: Checklists and Perspectives", 4th Asian Conference on Pattern Languages of Programs (APLoP), 2015

Hillside Group, "How to Hold a Writer's Workshop," and "Suggestions for a Successful Writer's Workshop," http://hillside.net/component/content/article/65howtorunplop/235howtoholdawritersworkshop

Richard P. Gabriel, "Writers' Workshops & The Work of Making Things," Addison Wesley, 2002.

James O. Coplien, "A Pattern Language for Writers' Workshops," in Pattern Languages of Program Design 4, Addison-Wesley, 2000

IEEE Std 610.12-1990 (R2002), "IEEE Standard Glossary of Software Engineering Terminology," 2002.

Keun Lee, "Development and Evaluation of Value-Based Review (VBR) Methods," VDM Verlag, 2008.

Thomas Thelin, Per Runeson and Claes Wohlin, "An Experimental Comparison of Usage-Based and Checklist-Based Reading," IEEE Transactions on Software Engineering, Vol.29, No.8, pp.687-704, 2003

Forrest Shull, "Software Reading Techniques," Encyclopedia of Software Engineering, John Wiley & Sons, 2002.

Forrest Shull, Jeffrey Carver, Guilherme H. Travassos, Jose Carlos Maldonado, Reidar Conradi, Victor R. Basili, "Replicated studies: building a body of knowledge about software reading techniques," Lecture notes on empirical software engineering, pp.39-84, 2003.

Christopher Alexander, "The Timeless Way of Building," Oxford University Press, New York, 1979.

Tiffany Winn and Paul Calder, "Is This a Pattern?," IEEE Software, Vol.19, No.1, pp.59-66, 2002.

Deepak Alur, John Crupi, Dan Malks, "Core J2EE Patterns: Best Practices and Design Strategies," Pearson Education, 2001.

Will Tracz, "RMISE Workshop on Software Reuse Meeting Summary," Software Reuse: Emerging Technology, IEEE CS, 1988.

Gerard Meszaros and Jim Doble, "A pattern language for pattern writing," Pattern languages of program design 3, pp.529-574, Addison-Wesley, 1997.

Neil B. Harrison, "The Language of Shepherding: A Pattern Language for Shepherds and Sheep," Pattern Languages of Program Design 5, Addison-Wesley, 2006.

ISO/IEC 25010: 2010 Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models, 2010.

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, 1998.

Hiroyuki Okamoto, et al., "An Experimental Evaluation of Individual Review Methods Focusing on Software Requirements Specifications Characteristics," SQiP Study Group Report, 2004. http://www.juse-sqip.jp/workshop/seika/2004/6/6_report.pdf

Eduardo B. Fernandez, Nobukazu Yoshioka, and Hironori Washizaki, "Two patterns for distributed systems: Enterprise Service Bus (ESB) and Distributed Publish/Subscribe," 18th Conference on Pattern Languages of Programs (PLoP), 2011.