Understanding Practitioners' Expectations on Clear Code Review Comments

理解从业者对清晰代码审查评论的期望

WARNING: This paper contains potentially offensive and harmful content.

警告：本文包含潜在的冒犯性和有害内容。

Abstract

摘要

The code review comment (CRC) is pivotal in the process of modern code review.

代码审查评论（CRC）在现代代码审查过程中至关重要。

It provides reviewers with the opportunity to identify potential bugs, offer constructive feedback, and suggest improvements.

它为审查者提供了识别潜在错误、提供建设性反馈和建议改进的机会。

Clear and concise code review comments (CRCs) facilitate the communication between developers and are crucial to the correct understanding of the identified issues and proposed solutions.

清晰且简洁的代码审查评论（CRC）促进了开发人员之间的沟通，对于正确理解已识别的问题和提议的解决方案至关重要。

Despite the importance of CRCs' clarity, there is still a lack of guidelines on what constitutes a good clarity and how to evaluate it.

尽管 CRC 的清晰度非常重要，但关于何为良好的清晰度以及如何对其进行评估，目前仍缺乏相关指南。

In this paper, we conduct a comprehensive study on understanding and evaluating the clarity of CRCs.

在本文中，我们进行了一项关于理解和评估 CRC 清晰度的综合研究。

We first derive a set of attributes related to the clarity of CRCs, namely RIE attributes (i.e., Relevance, Informativeness, and Expression), as well as their corresponding evaluation criteria based on our literature review and survey with practitioners.

我们首先根据文献综述和对从业者的调查，得出了一组与 CRC 清晰度相关的属性，即 RIE 属性（即相关性 Relevance、信息性 Informativeness 和表达 Expression），以及它们相应的评估标准。

We then investigate the clarity of CRCs in open-source projects written in nine programming languages and find that a large portion (i.e., 28.8%) of the CRCs lack the clarity in at least one of the attributes.

然后，我们调查了用九种编程语言编写的开源项目中的 CRC 清晰度，发现很大一部分（即 28.8%）的 CRC 在至少一个属性上缺乏清晰度。

Finally, we explore the potential of automatically evaluating the clarity of CRCs by proposing ClearCRC.

最后，我们通过提出 ClearCRC 来探索自动评估 CRC 清晰度的潜力。

Experimental results show that ClearCRC with pre-trained language models is promising for effective evaluation of the clarity of CRCs, achieving a balanced accuracy up to 73.04% and a F-1 score up to 94.61%.

实验结果表明，结合预训练语言模型的 ClearCRC 在有效评估 CRC 清晰度方面具有前景，实现了高达 73.04% 的平衡准确率和高达 94.61% 的 F-1 分数。

CCS Concepts: • Software and its engineering → Software creation and management.
CCS 概念：• 软件及其工程 → 软件创建与管理。

Additional Key Words and Phrases: Clarity, Code Review Comment, Code Review
附加关键词和短语：清晰度，代码审查评论，代码审查

1 Introduction
1 介绍

Code review is the process of systematic examinations on software source code performed by third-party developers [42, 49].
代码审查是由第三方开发人员对软件源代码进行系统检查的过程 [42, 49]。

The primary goals of code review include identifying potential issues, seeking areas for improvement, and transferring knowledge [9, 30, 51, 63, 65].
代码审查的主要目标包括识别潜在问题、寻求改进领域和传递知识 [9, 30, 51, 63, 65]。

It has been widely integrated into the software development life cycle in both open-source and industrial projects to help the assurance of software quality.
它已被广泛集成到开源项目和工业项目的软件开发生命周期中，以帮助保证软件质量。

A code review comment (CRC) is a specific piece of feedback provided by a reviewer during the code review process.
代码审查评论（CRC）是审查者在代码审查过程中提供的具体反馈意见。

Clear and concise code review comments (CRCs) are crucial for ensuring that the feedback is readable and actionable, and further contributing to the overall quality of the software.
清晰简洁的代码审查评论（CRC）对于确保反馈的可读性和可操作性至关重要，并进一步有助于提高软件的整体质量。

On the contrary, CRCs that lack of sufficient clarity may lead to confusion, misunderstandings, and misinterpretations amongst the collaborating developers.
相反，缺乏足够清晰度的 CRC 可能会导致协作开发人员之间的困惑、误解和曲解。

Figure 1 presents two examples of code changes and their corresponding code review comments.
图 1 展示了两个代码变更及其相应的代码审查评论的示例。

In Example 1, the reviewer comments “please revert this change”.
在示例 1 中，审查者评论道“请撤销此更改”。

However, no rationale or reason behind this comment is provided.
然而，该评论背后并未提供任何理由或原因。

Developers may not understand why this change needs to be reverted.
开发人员可能不理解为什么要撤销此更改。

In Example 2, the reviewer suggests renaming a parent class and also explains the reason of such suggestion.
在示例 2 中，审查者建议重命名父类，并解释了提出该建议的原因。

Moreover, this comment is written in a more friendly tone (i.e., “We can ...”).
此外，这条评论是用更友好的语气写的（即“我们可以……”）。

Although both of these two examples provide a suggestion to modify the code, their effectiveness in conveying reviewer’s idea may considerably vary.
尽管这两个例子都提供了修改代码的建议，但它们在传达审查者想法方面的有效性可能有很大差异。

Prior studies provide preliminary insights on revealing the quality of CRCs.
先前的研究为揭示 CRC 的质量提供了初步见解。

For example, usefulness [11, 27, 47] focuses on whether the CRCs can trigger subsequent code changes or if the reply to CRCs has a positive sentiment (e.g., “LGTM”).
例如，有用性 [11, 27, 47] 侧重于 CRC 是否能触发随后的代码更改，或者对 CRC 的回复是否具有积极的情感（例如，“LGTM”/看起来不错）。

Yang et al. [62] proposed four attributes to evaluate the quality of CRCs.
Yang 等人 [62] 提出了四个属性来评估 CRC 的质量。

Such attributes focus more on the purpose of CRCs (e.g., evaluation, suggestion, and question).
这些属性更多地关注 CRC 的目的（例如，评估、建议和提问）。

However, these studies either indirectly evaluate the quality of CRCs using the information after the completion of the review, or evaluate the CRCs based on whether it has elements related to the purpose of this comment.
然而，这些研究要么是利用审查完成后的信息间接评估 CRC 的质量，要么是基于 CRC 是否包含与其目的相关的要素来评估它。

Therefore, a systematic understanding and characterizing of how a CRC can clearly and concisely foster the communication among developers (i.e., clarity of CRCs) is still in mystery and an on-going challenge.
因此，系统地理解和表征 CRC 如何清晰简洁地促进开发人员之间的交流（即 CRC 的清晰度）仍然是一个谜，也是一个持续的挑战。

In this paper, we conduct a comprehensive study to uncover the clarity of CRCs by following a multi-phased investigation:
在本文中，我们通过多阶段调查进行了一项综合研究，以揭示 CRC 的清晰度：

1) we understand the characteristics and evaluation criteria of CRCs’ clarity through a systematic literature review, a preliminary review with industrial professionals, and an online questionnaire survey with practitioners;

1. 我们通过系统的文献综述、对行业专业人士的初步审查以及对从业者的在线问卷调查，了解 CRC 清晰度的特征和评估标准；

2) we examine the clarity of CRCs in open-source projects by conducting a manual investigation on sampled datasets;

2. 我们通过对抽样数据集进行人工调查，检查开源项目中 CRC 的清晰度；

3) we seek to automatically evaluate the clarity of CRCs by proposing an automated framework.

3. 我们试图通过提出一个自动化框架来自动评估 CRC 的清晰度。

Particularly, we study the clarity of code review comments by answering three research questions:
具体而言，我们通过回答三个研究问题来研究代码审查评论的清晰度：

RQ1: What attributes are relevant to the clarity of CRC?
RQ1：哪些属性与 CRC 的清晰度相关？

Based on the analysis on our literature review and 103 survey responses from practitioners, we derive our RIE attributes for the clarity of CRCs (i.e., Relevance, Informativeness, and Expression) and their corresponding evaluation criteria.
基于对文献综述和来自从业者的 103 份调查回复的分析，我们得出了 CRC 清晰度的 RIE 属性（即相关性 Relevance、信息性 Informativeness 和表达 Expression）及其相应的评估标准。

More than 75% of the participants consider that these attributes are important to the clarity of CRCs.
超过 75% 的参与者认为这些属性对 CRC 的清晰度很重要。

RQ2: How is the clarity of code review comments in open-source projects?
RQ2：开源项目中的代码审查评论的清晰度如何？

We manually investigate the clarity of code review comments in open-source projects using the datasets sampled from the work of Li et al. [33].
我们利用从 Li 等人 [33] 的工作中抽样的数据集，人工调查了开源项目中代码审查评论的清晰度。

We find that 28.8% of the CRCs in our study datasets are insufficient in at least one of the three attributes of CRCs' clarity.
我们发现，在我们的研究数据集中，28.8% 的 CRC 在 CRC 清晰度的三个属性中至少有一个方面存在不足。

Among these attributes, Informativeness has the most considerable insufficiency.
在这些属性中，信息性的不足最为显著。

RQ3: Can we automatically evaluate the clarity of code review comments?
RQ3：我们能否自动评估代码审查评论的清晰度？

We propose ClearCRC, an automated framework for the evaluation of CRCs' clarity based on the RIE attributes.
我们提出了 ClearCRC，这是一个基于 RIE 属性评估 CRC 清晰度的自动化框架。

We compare the results of ClearCRC with three sets of backbone models: 1) training deep learning and machine learning models; 2) fine-tuning pre-trained language models; and 3) prompting large language models (LLMs).
我们将 ClearCRC 的结果与三组骨干模型进行了比较：1）训练深度学习和机器学习模型；2）微调预训练语言模型；3）提示大型语言模型（LLMs）。

Our results show that ClearCRC is effective in evaluating each of the RIE attributes, with an average balanced accuracy of 73.04% using pre-trained language models.
我们的结果表明，ClearCRC 在评估每个 RIE 属性方面是有效的，使用预训练语言模型的平均平衡准确率为 73.04%。

We summarize the contributions of this paper as follows:
我们将本文的贡献总结如下：

• We derive RIE attributes and their corresponding evaluation criteria for the clarity of CRCs by analyzing the results of our literature review and 103 survey responses from practitioners around the world.
• 我们通过分析文献综述的结果和来自世界各地从业者的 103 份调查回复，推导出了 CRC 清晰度的 RIE 属性及其相应的评估标准。

• We find that a large portion of the CRCs in open source projects actually lack of sufficient clarity.
• 我们发现开源项目中很大一部分 CRC 实际上缺乏足够的清晰度。

We also publicly share our manually labelled data in the replication package [1] for future studies.
我们还在复制包 [1] 中公开分享了我们人工标记的数据，以供未来研究使用。

• We propose ClearCRC, an automated framework for the evaluation of CRCs' clarity using various backbone models such as machine learning, deep learning, pre-trained language models, and large language models.
• 我们提出了 ClearCRC，这是一个利用机器学习、深度学习、预训练语言模型和大型语言模型等各种骨干模型来评估 CRC 清晰度的自动化框架。

ClearCRC achieves promising results in our evaluation, especially using pre-trained language models.
ClearCRC 在我们的评估中取得了可喜的成果，尤其是使用预训练语言模型时。

Overall, the findings of our studies may be used as actionable guidelines for evaluating and writing clear CRCs, as well as for curating high-quality data to improve the automated generation techniques of CRCs.
总体而言，我们的研究结果可用作评估和编写清晰 CRC 的可操作指南，以及用于整理高质量数据以改进 CRC 的自动生成技术。

Paper Organization.
论文组织结构

Section 2 summarizes the related work.
第 2 节总结了相关工作。

Section 3 presents the methodology of our study.
第 3 节介绍了我们研究的方法。

Section 4 discusses the results of our research questions.
第 4 节讨论了我们研究问题的结果。

Section 5 discusses the implications of our study.
第 5 节讨论了我们研究的启示。

Section 6 discusses the threats to validity.
第 6 节讨论了对有效性的威胁。

Section 7 concludes the paper.
第 7 节总结了全文。

2 Related Work
2 相关工作

In this section, we summarize the related work in two aspects: studying the quality of code review comments and automated generation of code review comments.
在本节中，我们从两个方面总结相关工作：研究代码审查评论的质量和代码审查评论的自动生成。

2.1 Quality of Code Review Comments.
2.1 代码审查评论的质量。

Kerzazi et al. [7] found that sentiment conveyed within comments can significantly impact the outcome of the review process.
Kerzazi 等人 [7] 发现，评论中传达的情感会显著影响审查过程的结果。

Kononenko et al. [29] suggested that the review quality is mainly associated with the thoroughness of the feedback, the reviewer's familiarity with the code, and the perceived quality of the code itself.
Kononenko 等人 [29] 提出，审查质量主要与反馈的彻底性、审查者对代码的熟悉程度以及对代码本身质量的感知有关。

Rahman et al. [47] presented a comparative analysis of useful versus non-useful review comments, distinguishing them through their textual attributes and the reviewers' expertise.
Rahman 等人 [47] 对有用与无用的审查评论进行了比较分析，通过其文本属性和审查者的专业知识来区分它们。

Comments were classified as useful or non-useful depending on their capacity to instigate changes.
评论根据其引发变更的能力被分类为有用或无用。

Chouchen et al. [16] synthesized negative examples of code reviews that degraded software quality, categorizing erroneous practices into five patterns: Confused reviewers, Divergent reviewers, Low review participation, Shallow review, and Toxic review.
Chouchen 等人 [16] 综合了降低软件质量的负面代码审查示例，将错误做法归类为五种模式：困惑的审查者、分歧的审查者、低审查参与度、浅层审查和有毒审查。

Ram et al. [48] focused on the issue of code change reviewability, which is closely related to the quality of reviews.
Ram 等人 [48] 关注代码变更的可审查性问题，这与审查质量密切相关。

Ebert et al. [19, 20] identified confusion as a significant detriment to the quality of code reviews and offered recommendations for addressing issues of confusion.
Ebert 等人 [19, 20] 将困惑视为代码审查质量的重大损害，并提出了解决困惑问题的建议。

Bosu et al. [11] emphasized that the usefulness of code review lies in its ability to assist developers in avoiding defects, adhering to team conventions, and resolving issues efficiently and reasonably.
Bosu 等人 [11] 强调，代码审查的有用性在于它能够帮助开发人员避免缺陷、遵守团队惯例以及有效且合理地解决问题。

Ferreira et al. [23] highlighted the prevalence of uncivil behavior during the code review process, noting that discourteous comments can hinder project communication and discussion, ultimately slowing down development progress.
Ferreira 等人 [23] 强调了代码审查过程中不文明行为的普遍性，指出不礼貌的评论会阻碍项目沟通和讨论，最终拖慢开发进度。

Pascarella et al. [45] examined the essential information required by reviewers in code reviews, including the suitability of an alternative solution, correct understanding, rationale, code context, etc.
Pascarella 等人 [45] 检查了代码审查中审查者所需的基本信息，包括替代方案的适用性、正确理解、基本原理、代码上下文等。

Yang et al. [62] introduced four attributes for evaluating the quality of CRC: questions, suggestions, evaluations, and emotion, advocating for an assessment of the quality.
Yang 等人 [62] 介绍了评估 CRC 质量的四个属性：问题、建议、评估和情感，主张对质量进行评估。

Prior studies provide insights on the quality of CRCs from different perspectives.
先前的研究从不同角度提供了关于 CRC 质量的见解。

These studies generally emphasize the importance of “clear” CRCs.
这些研究普遍强调了“清晰”CRC 的重要性。

However, the understanding and characterizing on what are “clear” CRCs are still limited.
然而，对于什么是“清晰”CRC 的理解和表征仍然有限。

Therefore, in this paper, we conduct a comprehensive study to uncover and demystify what are the characteristics of a “clear” CRC.
因此，在本文中，我们进行了一项综合研究，以揭示和阐明“清晰”CRC 的特征是什么。

2.2 Automated Generation of Code Review Comments.
2.2 代码审查评论的自动生成。

The automated generation of code review comments has been widely studied in recent years [34, 40, 53, 58], aiming to streamline this critical yet often labor-intensive activity in the software engineering life cycle.
近年来，代码审查评论的自动生成得到了广泛研究 [34, 40, 53, 58]，旨在简化软件工程生命周期中这一关键但往往劳动密集型的活动。

Efforts in this domain can be categorized into three main types of approaches: traditional rule-based methods, deep learning techniques, and Large Language Models (LLMs) based techniques.
该领域的努力可归类为三种主要方法：传统的基于规则的方法、深度学习技术和基于大型语言模型（LLM）的技术。

Early automation efforts in code review were aimed at identifying code violations and defects utilizing traditional rule-based static analysis tools [8].
早期的代码审查自动化工作旨在利用传统的基于规则的静态分析工具识别代码违规和缺陷 [8]。

While providing a base for automation, they lacked the flexibility to adapt to the nuanced and evolving nature of software development practices.
虽然为自动化提供了基础，但它们缺乏适应软件开发实践细微差别和不断演变性质的灵活性。

The emergence of deep learning has significantly enhanced the capability to automate code reviews, offering a nuanced understanding and interpretation of code changes.
深度学习的出现显著增强了自动化代码审查的能力，提供了对代码变更的细致理解和解释。

Techniques leveraging LSTM [26], and Transformers [34, 57] have been pivotal in predicting review necessities and generating context-specific feedback.
利用 LSTM [26] 和 Transformer [34, 57] 的技术在预测审查必要性和生成特定于上下文的反馈方面发挥了关键作用。

Li et al. [34] proposed a pre-trained model based on the Text-To-Text-Transfer Transformer (T5) model [46], specifically tailored for the code review process across three different code review tasks including the generation of CRCs.
Li 等人 [34] 提出了一种基于文本到文本传输变换器（T5）模型 [46] 的预训练模型，专门针对包括 CRC 生成在内的三个不同代码审查任务的代码审查过程进行了定制。

LLaMA-Reviewer [40] utilized the LLaMA model and parameter-efficient fine-tuning to automate code review comments generation, achieving performance on par with existing code-review-focused models using fewer resources.
LLaMA-Reviewer [40] 利用 LLaMA 模型和参数高效微调来自动化代码审查评论生成，使用更少的资源实现了与现有专注于代码审查的模型相当的性能。

Prior studies commonly incorporated the CRCs data directly, without a curation or quality selection process.
先前的研究通常直接纳入 CRC 数据，没有进行整理或质量筛选过程。

Given this scenario, existing CRCs generation techniques might inadvertently learn from CRCs data lacking clarity, resulting in perplexing outcomes.
鉴于这种情况，现有的 CRC 生成技术可能会无意中从缺乏清晰度的 CRC 数据中学习，导致令人困惑的结果。

Our study can complement the research of automated CRCs generation to curate the training data, and further improve the quality of the generated CRCs.
我们的研究可以补充自动 CRC 生成的研究，以整理训练数据，并进一步提高生成的 CRC 的质量。

3 Methodology
3 方法论

Figure 2 presents an overview of our study.
图 2 展示了我们研究的概览。

To address the three research questions proposed in the Introduction, we conduct a comprehensive study that involves three phases.
为了解决引言中提出的三个研究问题，我们要进行一项涉及三个阶段的综合研究。

Phase 1: We first derive an initial list of attributes and evaluation criteria concerning the clarity of CRCs through literature review and a preliminary review with industrial professionals.
第一阶段：我们首先通过文献综述和与行业专业人士的初步审查，得出一份关于 CRC 清晰度的属性和评估标准的初步清单。

We then survey with practitioners for their perspectives to refine the attributes and evaluation criteria.
然后，我们调查从业者的观点，以完善属性和评估标准。

Phase 2: We manually investigate the clarity of CRCs in open-source projects using the clarity attributes and evaluation criteria derived in the prior phase.
第二阶段：我们使用前一阶段得出的清晰度属性和评估标准，手动调查开源项目中 CRC 的清晰度。

Phase 3: We propose ClearCRC, an automated framework that seeks to evaluate the clarity of CRCs based on the attributes derived from our literature review and practitioners' feedback.
第三阶段：我们提出 ClearCRC，这是一个自动化框架，旨在根据我们从文献综述和从业者反馈中得出的属性来评估 CRC 的清晰度。

3.1 Characterizing the Clarity of CRCs
3.1 表征 CRC 的清晰度

We characterize the attributes related to the clarity of CRCs by combining 1) a systematic literature review followed by a preliminary review with industrial professionals, and 2) the analysis of our online survey with practitioners.
我们通过结合 1）系统的文献综述以及随后的行业专业人士初步审查，和 2）对从业者在线调查的分析，来表征与 CRC 清晰度相关的属性。

3.1.1 Literature Review. We analyze existing studies on the quality of CRCs and derive an initial set of attributes related to the clarity of CRCs.
3.1.1 文献综述。我们分析了关于 CRC 质量的现有研究，并得出了一组与 CRC 清晰度相关的初步属性。

Literature Collection. We first gather the papers related to code review by utilizing the list of papers provided by prior literature reviews [17, 60].
文献收集。我们首先利用先前的文献综述 [17, 60] 提供的论文列表，收集与代码审查相关的论文。

These two literature reviews summarized 139 and 112 prior works on code review published from 2005 to 2019 and 2011 to 2019, respectively.
这两篇文献综述分别总结了 2005 年至 2019 年和 2011 年至 2019 年发表的 139 篇和 112 篇关于代码审查的先前著作。

We then follow the strategy of paper collection in these literature reviews to collect papers published after 2019 and collect 70 papers in this process.
然后，我们遵循这些文献综述中的论文收集策略，收集 2019 年之后发表的论文，并在此过程中收集了 70 篇论文。

Data Analysis. We first read the titles and abstracts of all the collected papers and filter papers that are not related to the studies on CRCs.
数据分析。我们首先阅读所有收集到的论文的标题和摘要，并过滤掉与 CRC 研究无关的论文。

The topics of such filtered papers include studying the code change, authors, reviewers, pull requests, and commit messages.
此类被过滤论文的主题包括研究代码变更、作者、审查者、拉取请求和提交信息。

After this process, we have a list of 47 papers that are related to CRCs for further analysis.
经过这一过程，我们得到了一份包含 47 篇与 CRC 相关的论文列表，以供进一步分析。

Two authors of this paper then independently read these papers and generate an initial set of codes related to the attribute of CRCs' clarity.
本文的两位作者随后独立阅读这些论文，并生成一组与 CRC 清晰度属性相关的初始代码。

The authors then perform open card sorting [54] on the generated codes to analyze the codes and sort the generated codes into potential themes that indicate the attributes related to the clarity of CRCs.
然后，作者对生成的代码执行开放式卡片分类 [54]，以分析代码并将生成的代码分类为潜在主题，这些主题指示了与 CRC 清晰度相关的属性。

Particularly, author 1 generates six initial codes: Confused Reviews, Toxic Reviews, Relevant Reviews, Readable Reviews, Shallow Reviews, and Informative Reviews.
具体而言，作者 1 生成了六个初始代码：困惑的审查、有毒的审查、相关的审查、可读的审查、浅层的审查和信息丰富的审查。

Author 2 generates five initial codes: Readability, Sentiment, Relevance, Information, and Toxicity.
作者 2 生成了五个初始代码：可读性、情感、相关性、信息和毒性。

The two authors then engage in thorough discussions to refine the attribute set.
随后，两位作者进行了深入讨论，以完善属性集。

Both authors collaboratively re-evaluate the attributes and reconcile differences by focusing on semantic consistency and conceptual clarity.
两位作者通过关注语义一致性和概念清晰度，共同重新评估属性并协调差异。

Throughout this process, attributes that exhibit overlap or ambiguity are merged or redefined.
在整个过程中，表现出重叠或歧义的属性被合并或重新定义。

For example, Readability and Toxicity are merged into Expression, and the meaning of Shallow Reviews can be represented by Informativeness.
例如，可读性和毒性被合并为表达，浅层审查的含义可以用信息性来表示。

Eventually, we derive three attributes that are related to the clarity of CRCs, including Relevance, Informativeness, and Expression.
最终，我们得出了三个与 CRC 清晰度相关的属性，包括相关性、信息性和表达。

The authors then discuss and generate an initial definition for each attribute.
然后，作者讨论并为每个属性生成初始定义。

Preliminary Review. To preliminarily verify the attributes derived from our literature review, we conduct a group interview with 11 industrial practitioners to obtain their feedback.
初步审查。为了初步验证我们从文献综述中得出的属性，我们与 11 位行业从业者进行了小组访谈以获取他们的反馈。

The participants are full-time engineers at Company X, which is a world-leading IT company.
参与者是 X 公司的全职工程师，该公司是一家世界领先的 IT 公司。

All of the participants have at least three years of experience in code review and software development.
所有参与者都拥有至少三年的代码审查和软件开发经验。

The duration of this group interview is around 1 hour.
这次小组访谈的时间约为 1 小时。

We follow a three-step process to conduct the group interview:
我们遵循三个步骤来进行小组访谈：

1) We ask the participants to freely talk about their expectations on the clarity of CRCs without the knowledge of our derived attributes;

1. 我们要求参与者在不了解我们推导出的属性的情况下，自由谈论他们对 CRC 清晰度的期望；

2) We present our derived attributes to the participants;

2. 我们向参与者展示我们推导出的属性；

3) We discuss with the participants for whether the attributes can reflect their expectations on the clarity of CRCs.

3. 我们与参与者讨论这些属性是否能反映他们对 CRC 清晰度的期望。

Eventually, we find that most of the points initially proposed by the participants can be reflected by our derived attributes.
最终，我们发现参与者最初提出的大部分观点都可以通过我们推导出的属性反映出来。

The remaining non-reflected points are related to the process of code review rather than the code review comments, including deciding proper reviewers and prompt response time.
其余未反映的观点与代码审查的过程有关，而不是与代码审查评论有关，包括决定合适的审查者和及时的响应时间。

At the end, the participants are thanked and briefly informed about the next plans.
最后，我们感谢了参与者，并简要告知了接下来的计划。

The participants also provide suggestions regarding the design of surveys.
参与者还就调查的设计提供了建议。

We take their suggestions into consideration while designing our survey in the next step.
我们在设计下一步的调查时，考虑了他们的建议。

3.1.2 Questionnaire Survey with Practitioners.
3.1.2 从业者问卷调查。

We conduct an online questionnaire survey with practitioners for their perspectives on the clarity of CRCs and further refine the attributes and derive the evaluation criteria.
我们对从业者进行在线问卷调查，以了解他们对 CRC 清晰度的看法，并进一步完善属性并得出评估标准。

Survey Design.
调查设计。

The questions in our survey are divided into three parts:
我们调查中的问题分为三个部分：

P1 We ask the participants for their basic information (e.g., country or region of residence, primary job role, and years of experience in the primary job role) and if they have experience in code review.
P1 我们询问参与者的基本信息（例如，居住国家或地区、主要工作职责以及主要工作职责的从业年限），以及他们是否有代码审查经验。

P2 For each attribute of the clarity, we ask the participants for “To what extent do you think that the aspect of "{attribute}" is important to the clarity of code review comments?” where {attribute} is replaced with a specific one.
P2 对于清晰度的每个属性，我们询问参与者“您认为“{attribute}”这方面对代码审查评论的清晰度有多重要？”，其中 {attribute} 替换为具体的属性。

The participants then choose from “Very important”, “Important”, “Neutral”, “Unimportant”, and “Very unimportant”.
然后，参与者从“非常重要”、“重要”、“中立”、“不重要”和“非常不重要”中进行选择。

If “Very important” or “important” is chosen, we further ask the participants for the details of how they will evaluate the corresponding attribute (e.g., what factors and detailed information they may focus on).
如果选择“非常重要”或“重要”，我们会进一步询问参与者如何评估相应属性的细节（例如，他们可能关注哪些因素和详细信息）。

We will derive the evaluation criteria based on their comments.
我们将根据他们的评论得出评估标准。

If other options are chosen, we ask the participant for the potential reasons.
如果选择其他选项，我们会询问参与者潜在的原因。

At the end of this part, we further ask the participant for “Do you have some ideas about other important aspects that contribute to the clarity of code review comments?”.
在这一部分的最后，我们进一步询问参与者“您是否认为还有其他重要方面有助于代码审查评论的清晰度？”。

P3 We ask the participants for questions on automated code review comment generation, such as their experience in using such tools, and their perspectives on the clarity of the generated CRCs.
P3 我们向参与者询问有关自动代码审查评论生成的问题，例如他们使用此类工具的经验，以及他们对生成的 CRC 清晰度的看法。

Survey Implementation.
调查实施。

We implement the survey following the design discussed above using Microsoft Forms [4].
我们使用 Microsoft Forms [4] 按照上述设计实施了调查。

We conduct a pilot survey with three practitioners to collect their feedback on the design of our survey.
我们对三位从业者进行了一次试点调查，以收集他们对我们调查设计的反馈。

All the practitioners in the pilot survey have experience in writing and reviewing CRCs.
试点调查中的所有从业者都有撰写和审查 CRC 的经验。

The pilot participants provide suggestions regarding the clarification of instructions and the consistency of some terms.
试点参与者就说明的澄清和一些术语的一致性提供了建议。

We make modifications according to their feedback and have a final version of the survey, which is an anonymous questionnaire and can be accessed using the link provided in our email sent to the participants.
我们根据他们的反馈进行了修改，得到了最终版本的调查问卷，这是一个匿名问卷，可以通过我们发送给参与者的电子邮件中提供的链接访问。

A sample of the complete survey is available in our replication package [1].
完整的调查样本可在我们的复制包 [1] 中找到。

Participants.
参与者。

To invite participants from diverse backgrounds, we reach out to industrial and academic professionals residing in various countries or regions, across five continents around the world.
为了邀请来自不同背景的参与者，我们联系了居住在世界五大洲各个国家或地区的行业和学术专业人士。

Eventually, we receive 112 responses in total.
最终，我们总共收到了 112 份回复。

We then filter 9 responses of which indicate as having no experience in code review, resulting in 103 remaining responses for further analysis.
然后，我们过滤掉 9 份表明没有代码审查经验的回复，剩下 103 份回复用于进一步分析。

➤ Demographics.
➤ 人口统计。

Table 1 shows the statistical information of the participants.
表 1 显示了参与者的统计信息。

The participants reside in 37 countries or regions across five continents, including 49 participants in Europe, 22 participants in Asia, 24 participants in North America, 4 participants in South America, and 4 participants in Oceania.
参与者居住在五大洲的 37 个国家或地区，其中欧洲 49 人，亚洲 22 人，北美洲 24 人，南美洲 4 人，大洋洲 4 人。

A majority of the participants have an occupation of industrial/freelance professional (76.7%) and primary job role as development (81.6%).
大多数参与者的职业是工业/自由职业专业人士（76.7%），主要工作职责是开发（81.6%）。

A large percentage of the participants (68.9%) have at least five years of experience in their primary job role.
很大一部分参与者（68.9%）在主要工作岗位上拥有至少五年的经验。

In short, our survey participants reside in various countries or regions all over the world, and most of them are industrial or freelance professionals with more than 5 years of experience in software development.
简而言之，我们的调查参与者居住在世界各地的各个国家或地区，其中大多数是拥有超过 5 年软件开发经验的工业或自由职业专业人士。

Data Analysis.
数据分析。

The data we obtain from the survey consists of option data from multiple-choice questions and natural language response data from open-ended questions.
我们从调查中获得的数据包括来自多项选择题的选项数据和来自开放式问题的自然语言回答数据。

(1) For the questions of multiple choices, we compute the percentage of each option, e.g., the percentage of survey participants who think "Relevance" is "Very Important" to the clarity of CRCs shown in Fig. 3.
(1) 对于多项选择题，我们计算每个选项的百分比，例如，如图 3 所示，认为“相关性”对 CRC 清晰度“非常重要”的调查参与者百分比。

(2) For the open questions (e.g., details for evaluating the attributes), we generate codes from the answers and perform open card sorting [54] to analyze the thematic similarity.
(2) 对于开放式问题（例如，评估属性的细节），我们从答案中生成代码，并执行开放式卡片分类 [54] 以分析主题相似性。

Specifically, to derive the evaluation criteria for each attribute, we first extract and record criteria of all the responses, and then sort and categorize them into concise and explicit descriptions like "Proper syntax and grammar".
具体来说，为了推导出每个属性的评估标准，我们首先提取并记录所有回答的标准，然后将它们分类归纳为简洁明了的描述，如“正确的语法和文法”。

For example, a response from one survey participant for Informativeness said “Explain why, with specific reference to the change.” will finally lead to two criteria including “I.E2: Provide reasons or context information” and “I.O2: Provide reference information” (See Section 4).
例如，一位调查参与者对“信息性”的回答是“解释原因，并具体参考该变更。”最终将引出两个标准，包

括“I.E2：提供原因或背景信息”和“I.O2：提供参考信息”（见第 4 节）。

When a consensus on these criteria for each attribute is reached, we first filter evaluation criteria mentioned fewer than five times (i.e., 1-4 times), and then select evaluation criteria which are mentioned 15+ times as essential evaluation criteria and the remaining ones as optional evaluation criteria.
当就每个属性的这些标准达成共识后，我们首先过滤掉提及次数少于 5 次（即 1-4 次）的评估标准，然后选择提及次数在 15 次以上的评估标准作为*基本*评估标准，其余的作为*可选*评估标准。

We take them as references for manual investigation.
我们将它们作为人工调查的参考。

Detailed results will be presented in Section 4 (RQ1).
详细结果将在第 4 节（RQ1）中呈现。

3.2 Investigating the Clarity of CRCs in Open-Source Projects
3.2 调查开源项目中 CRC 的清晰度

In this phase, we manually investigate the clarity of CRCs in open-source projects using the attributes and evaluation criteria of clarity derived in the prior phase.
在这一阶段，我们使用前一阶段得出的清晰度属性和评估标准，手动调查开源项目中 CRC 的清晰度。

3.2.1 Data Preparation.
3.2.1 数据准备。

We use the benchmark dataset proposed by Li et al. [33] to conduct manual investigation.
我们使用 Li 等人 [33] 提出的基准数据集来进行人工调查。

The dataset contains pairs of diff hunk and CRC written in nine programming languages.
该数据集包含用九种编程语言编写的 diff hunk（代码差异片段）和 CRC 对。

Specifically, we randomly sample a set of data from its validation dataset for each programming language.
具体来说，我们从每种编程语言的验证数据集中随机抽取一组数据。

We do not sample from its training dataset because the data in different programming language is combined together and can not be distinguished.
我们不从其训练数据集中抽样，因为不同编程语言的数据是混合在一起的，无法区分。

For each programming language, we randomly sample a set of data based on 95% confidence level and 5% confidence interval [10].
对于每种编程语言，我们基于 95% 的置信水平和 5% 的置信区间随机抽取一组数据 [10]。

Table 2 presents the details of our sampled datasets.
表 2 展示了我们抽样数据集的详细信息。

In total, we randomly sample 2,438 pairs of diff hunk and CRC.
我们总共随机抽取了 2,438 对 diff hunk 和 CRC。

The sample size of each programming language varies from 216 for C to 339 for Golang.
每种编程语言的样本量从 C 语言的 216 个到 Golang 的 339 个不等。

3.2.2 Manual Investigation.
3.2.2 人工调查。

Two authors of this paper first carefully read the attributes and evaluation criteria derived in the previous phase, and discuss until the two authors have a clear and consistent understanding on the details.
本文的两位作者首先仔细阅读了前一阶段得出的属性和评估标准，并进行讨论，直到两位作者对细节有清晰一致的理解。

For each sampled data, the two authors then independently examine the CRC and its corresponding code change to label if it meets the evaluation criteria for each attribute.
对于每个抽样数据，两位作者随后独立检查 CRC 及其相应的代码变更，以标记其是否符合每个属性的评估标准。

When the process of labelling is completed, the two authors compare their results and discuss each disagreement until reaching a consensus.
当标记过程完成后，两位作者比较他们的结果并讨论每一个分歧，直到达成共识。

We have a Cohen’s Kappa [41] value of 0.87 in this process, which indicates a substantial agreement.
在此过程中，我们的 Cohen’s Kappa [41] 值为 0.87，这表明具有很高的一致性。

We will discuss the results of our manual investigate in Section 4 (RQ2).
我们将在第 4 节（RQ2）中讨论我们人工调查的结果。

3.3 ClearCRC: Automatically Evaluating the Clarity of CRCs
3.3 ClearCRC：自动评估 CRC 的清晰度

In this phase, we propose ClearCRC, an automated framework that aims at the evaluation of the clarity of CRCs, based on the RIE attributes derived from our literature review and practitioners’ feedback.
在这一阶段，我们提出了 ClearCRC，这是一个自动化框架，旨在基于我们从文献综述和从业者反馈中得出的 RIE 属性来评估 CRC 的清晰度。

We adopt different sets of backbone models in our framework to empirically study their effectiveness in automatically evaluating the clarity of CRCs.
我们在框架中采用了不同组的骨干模型，以实证研究它们在自动评估 CRC 清晰度方面的有效性。

3.3.1 Models. We use three sets of backbone models, including deep learning and machine learning models, pre-trained language models (e.g., CodeBERT [22] and CodeReviewer [33]), and large language models (e.g., Llama [56] and CodeLlama [50]).
3.3.1 模型。我们使用了三组骨干模型，包括深度学习和机器学习模型、预训练语言模型（例如 CodeBERT [22] 和 CodeReviewer [33]）以及大型语言模型（例如 Llama [56] 和 CodeLlama [50]）。

Model Set 1: Deep Learning and Machine Learning Models. Prior studies on classifying good commit messages [55] and log messages [31] indicate that Bi-LSTM and Random Forest are effective in such classifications.
模型组 1：深度学习和机器学习模型。先前关于分类良好提交信息 [55] 和日志信息 [31] 的研究表明，Bi-LSTM 和随机森林在此类分类中是有效的。

Following these studies, we use Bi-LSTM [52] and Random Forest [12] as the deep learning and machine learning based backbones to perform the evaluation of CRCs' clarity.
遵循这些研究，我们使用 Bi-LSTM [52] 和随机森林 [12] 作为基于深度学习和机器学习的骨干网络，来执行 CRC 清晰度的评估。

Model Set 2: Pre-trained Language Models. For pre-trained language models, we use CodeBERT [22] and CodeReviewer [34] as our subject techniques.
模型组 2：预训练语言模型。对于预训练语言模型，我们使用 CodeBERT [22] 和 CodeReviewer [34] 作为我们的目标技术。

CodeBERT is a bimodal pre-trained language model for programming languages and natural languages with the same model architecture as RoBERTa-base [38].
CodeBERT 是一种用于编程语言和自然语言的双模态预训练语言模型，具有与 RoBERTa-base [38] 相同的模型架构。

It has been widely used by prior studies for classification tasks and presents a promising balance between performance and cost of computing resources [59, 66, 67].
它已被先前的研究广泛用于分类任务，并在性能和计算资源成本之间展现了良好的平衡 [59, 66, 67]。

CodeReviewer is a pre-trained model specialized for the automation of code review activities.
CodeReviewer 是一种专门用于自动化代码审查活动的预训练模型。

They proposed pre-training tasks designed for code review like code diff denoising, and then pre-trained the CodeT5 [61] model on a large-scale code review dataset.
他们提出了为代码审查设计的预训练任务，如代码差异去噪，然后在大规模代码审查数据集上预训练了 CodeT5 [61] 模型。

CodeReviewer shows competitive performance on code review tasks such as code refinement.
CodeReviewer 在代码改进等代码审查任务上表现出具有竞争力的性能。

Model Set 3: Large Language Models. LLMs have demonstrated promising results in various software engineering tasks [13, 15, 37], which brings opportunities and challenges for using LLMs as evaluators [14, 36].
模型组 3：大型语言模型。LLM 在各种软件工程任务中展示了可喜的成果 [13, 15, 37]，这为使用 LLM 作为评估者带来了机遇和挑战 [14, 36]。

For LLM baselines, we use Llama3-70B-Instruct [56] and CodeLlama-34B-Instruct [50].
作为 LLM 基线，我们使用 Llama3-70B-Instruct [56] 和 CodeLlama-34B-Instruct [50]。

We choose them since Llama series models show great performance among different LLMs [50, 56], and they are very popular in research related to code review [40, 64, 68].
我们选择它们是因为 Llama 系列模型在不同 LLM 中表现出色 [50, 56]，并且它们在与代码审查相关的研究中非常受欢迎 [40, 64, 68]。

Additionally, we also attempt to include a code-review-specialized LLM named LLaMA-Reviewer [40].
此外，我们还尝试包含一个名为 LLaMA-Reviewer [40] 的代码审查专用 LLM。

However, since LLaMA-Reviewer is tailored to specific downstream tasks and does not generalize well to our setting (i.e., it tends to generate invalid outputs when prompted), we finally decide to exclude it.
然而，由于 LLaMA-Reviewer 是针对特定下游任务定制的，不能很好地泛化到我们的设置中（即在提示时往往会生成无效输出），我们最终决定将其排除。

3.3.2 Data. Here we introduce the datasets we use as well as their augmentation and pre-processing.
3.3.2 数据。这里我们介绍我们使用的数据集及其增强和预处理。

Datasets and Augmentation. We utilize the manually labelled datasets in the prior phase to conduct the study, which consists of 2,438 pairs of code change and CRC in total.
数据集与增强。我们利用前一阶段人工标记的数据集进行研究，该数据集总共包含 2,438 对代码变更和 CRC。

We randomly split the datasets into 80% training, 10% validation, and 10% testing.
我们将数据集随机分为 80% 训练集、10% 验证集和 10% 测试集。

As shown in Table 4, the distribution of negative and positive instances is imbalanced in the dataset.
如表 4 所示，数据集中负例和正例的分布是不平衡的。

To mitigate such impact, for each experiment, we perform up-sampling on the corresponding attribute.
为了减轻这种影响，对于每个实验，我们对相应的属性执行上采样。

Specifically, we randomly repeat the negative instances of the experimented attribute in the training dataset to have the same amount as the positive instances.
具体来说，我们在训练数据集中随机重复实验属性的负例，使其数量与正例相同。

Note that we only augment data in the training set and ensure the testing set is consistent for all backbone models.
请注意，我们仅增强训练集中的数据，并确保所有骨干模型的测试集是一致的。

Processing. We analyze the raw input data including pairs of code change and CRC, process and combine the data with code change and CRC to feed into the model.
处理。我们分析包括代码变更和 CRC 对在内的原始输入数据，处理并结合代码变更和 CRC 数据以输入模型。

We remove the lines of code that are unrelated to the code changes (e.g., the surrounding code of code changes).
我们删除了与代码变更无关的代码行（例如，代码变更周围的代码）。

For models in set 1&2, we replace the “-” and “+” mark at the start of each line with “[DELETE]” and “[ADD]” in the code change, respectively.
对于组 1 和组 2 中的模型，我们分别将代码变更中每行开头的“-”和“+”标记替换为“[DELETE]”和“[ADD]”。

We then concatenate the CRC and the processed code change together, and attach a “[SEP]” token between them.
然后我们将 CRC 和处理后的代码变更连接在一起，并在它们之间附加一个“[SEP]”标记。

For large language models in set 3, We embed the information of code change and CRC into the prompt and inference the models to obtain the results returned by the models and further evaluate the clarity of each attribute.
对于组 3 中的大型语言模型，我们将代码变更和 CRC 的信息嵌入到提示中，并推理模型以获得模型返回的结果，从而进一步评估每个属性的清晰度。

For the prompt of using LLMs, we follow Prompt Engineering Guide to design the prompts [6].
对于使用 LLM 的提示，我们遵循提示工程指南来设计提示 [6]。

As shown in Figure 5, we first provide an instruction of the task, the attributes, and evaluation criteria.
如图 5 所示，我们首先提供任务、属性和评估标准的说明。

We then inform the models of how to use the evaluation criteria (i.e., meet all of the essential ones and at least one of the optional ones) and the expected template of output.
然后我们告知模型如何使用评估标准（即满足所有基本标准和至少一个可选标准）以及预期的输出模板。

Finally, we attach the actual data (i.e., diff hunk and CRC) and have the model start its evaluation.
最后，我们附上实际数据（即 diff hunk 和 CRC）并让模型开始评估。

3.3.3 Evaluation. We introduce the evaluation details in our empirical study.
3.3.3 评估。我们在实证研究中介绍评估细节。

Metrics. We use four metrics to evaluate the results of ClearCRC and the baselines: 1) balanced accuracy, 2) precision, 3) recall, and 4) F-1 score.
指标。我们使用四个指标来评估 ClearCRC 和基线的结果：1）平衡准确率，2）精确率，3）召回率，4）F-1 分数。

Balanced accuracy is computed based on the average of true positive rate and true negative rate.
平衡准确率是基于真阳性率和真阴性率的平均值计算的。

A higher balanced accuracy indicates a better capability in identifying both positive and negative instances.
较高的平衡准确率表明在识别正例和负例方面具有更好的能力。

The balanced accuracy of random guess in binary classification is close to 50.0% [32].
二分类中随机猜测的平衡准确率接近 50.0% [32]。

It is widely used by prior work to evaluate the performance of binary classification, especially on imbalanced data [32, 69].
它被先前的工作广泛用于评估二分类的性能，尤其是在不平衡数据上 [32, 69]。

For the calculation of precision, recall, and F-1 score which focus on the classification performance of positive instances, we consider CRCs that meet the evaluation criteria as positive instances, and otherwise as negative instances.
对于侧重于正例分类性能的精确率、召回率和 F-1 分数的计算，我们将符合评估标准的 CRC 视为正例，否则视为负例。

K-Fold Cross Validation. We utilize 5-fold cross validation to mitigate the impact of randomness, where 5 is a commonly used K value in prior studies involving k-fold cross validation [24, 28, 43].
K 折交叉验证。我们利用 5 折交叉验证来减轻随机性的影响，其中 5 是涉及 K 折交叉验证的先前研究中常用的 K 值 [24, 28, 43]。

We randomly split the dataset into five subsets.
我们将数据集随机分为五个子集。

The validation has five rounds in total.
验证总共有五轮。

For each round of validation, we use one subset (i.e., 20%) for validation and testing (i.e., half for validation and half for testing), and the remaining four subsets (i.e., 80%) for training.
对于每一轮验证，我们使用一个子集（即 20%）进行验证和测试（即一半用于验证，一半用于测试），其余四个子集（即 80%）用于训练。

We ensure that the dataset for each fold is identical for all models to perform a fair comparison.
我们确保每个折叠的数据集对所有模型都是相同的，以进行公平比较。

3.3.4 Implementation Details.
3.3.4 实现细节。

(1) For models in set 1, we use PyTorch [2] to implement Bi-LSTM and use Scikit-learn [3] to implement Random Forests, respectively.
(1) 对于组 1 中的模型，我们分别使用 PyTorch [2] 实现 Bi-LSTM，使用 Scikit-learn [3] 实现随机森林。

We follow prior studies [31, 55] to set the hyperparameters of the networks and the training processes.
我们遵循先前的研究 [31, 55] 来设置网络的超参数和训练过程。

(2) For pre-trained models in set 2, we access the models through the official checkpoints released on HuggingFace [5].
(2) 对于组 2 中的预训练模型，我们通过 HuggingFace [5] 上发布的官方检查点访问模型。

As to hyperparameters like batch size and learning rate, we tune them according to the official replication package and the volume of our datasets.
关于批量大小和学习率等超参数，我们根据官方复制包和我们数据集的体量进行调整。

During the training stage, we set the number of training epochs to 10 and perform the strategy of early stopping (n=3) on all models to limit the training consumption and save the best-performing model on the validation dataset for further testing.
在训练阶段，我们将训练轮数设置为 10，并对所有模型执行早停策略（n=3），以限制训练消耗并在验证数据集上保存表现最佳的模型以供进一步测试。

We adopt the AdamW [39] optimizer and linear scheduler.
我们采用 AdamW [39] 优化器和线性调度器。

(3) For large language models, we download their official checkpoints from HuggingFace.
(3) 对于大型语言模型，我们从 HuggingFace 下载其官方检查点。

We set the maximum number of generated tokens to 32 which is appropriate for the output format, and keep the other parameters the same in the default configuration.
我们将最大生成令牌数设置为 32，这对于输出格式是合适的，并保持其他参数与默认配置相同。

We publicly release the scripts, parameters, and datasets for further research [1].
我们公开发布了脚本、参数和数据集以供进一步研究 [1]。

4 Results
4 结果

In this section, we discuss the results of our RQs.
在本节中，我们讨论我们研究问题的结果。

4.1 RQ1: Characterizing and Understanding the Clarity of CRCs
4.1 RQ1：表征和理解 CRC 的清晰度

In this RQ, we discuss the results of our RIE attributes and evaluation criteria of CRCs' clarity, derived from our literature review and practitioners' survey.
在这个 RQ 中，我们讨论根据文献综述和从业者调查得出的 RIE 属性和 CRC 清晰度评估标准的结果。

Table 3 shows an overview of the attributes and their evaluation criteria.
表 3 显示了属性及其评估标准的概述。

Below, for each attribute, we discuss its detailed evaluation criteria and our survey results.
下面，对于每个属性，我们讨论其详细的评估标准和我们的调查结果。

4.1.1 Relevance. If the code review comment is relevant to the code change.
4.1.1 相关性。代码审查评论是否与代码变更相关。

Evaluation Criteria. After data analysis of open questions in our survey, we derive one essential (i.e., R.E1) and two optional (i.e., R.O1 and R.O2) evaluation criteria corresponding to the attribute of relevance.
评估标准。通过对我们调查中开放式问题的数据分析，我们得出了一项基本（即 R.E1）和两项可选（即 R.O1 和 R.O2）的与相关性属性对应的评估标准。

Figure 4 shows the frequency of evaluation criteria mentioned by our survey participants.
图 4 显示了我们的调查参与者提到的评估标准的频率。

R.E1 Relevant to the code change. The CRC should be self-explanatory and relevant to the code change.
R.E1 与代码变更相关。CRC 应当是自解释的，并且与代码变更相关。

It can be interpreted based on the current code change without a relying relevance to external information (e.g., other CRCs).
它可以基于当前的代码变更进行解释，而不依赖于与外部信息（例如其他 CRC）的相关性。

R.O1 Specify the relevant location. The CRC specifies the particular position of the code which has the issues or concerns.
R.O1 指定相关位置。CRC 指定了存在问题或疑虑的代码的具体位置。

R.O2 Correctly understand the code change. The CRC explicitly shows that the reviewer correctly understands the code change.
R.O2 正确理解代码变更。CRC 明确显示审查者正确理解了代码变更。

Discussion. Figure 3 shows the percentage of each rate given by the survey participants regarding the importance of each attribute.
讨论。图 3 显示了调查参与者对每个属性重要性给出的评分百分比。

Overall, most of the participants consider Relevance is important to the clarity of CRCs, including 67.0% as very important and 23.3% as somewhat important.
总体而言，大多数参与者认为相关性对 CRC 的清晰度很重要，其中 67.0% 认为非常重要，23.3% 认为比较重要。

Below, we present the comments from our survey participants regarding their perspectives on the evaluation criteria of each attribute.
下面，我们展示调查参与者关于他们对每个属性评估标准的看法的评论。

We correspond such comments to the evaluation criteria discussed above, and the corresponding part is highlighted in bold.
我们将这些评论与上述讨论的评估标准相对应，相应部分以粗体显示。

R.E1 "If the comment is made about the intent or the change itself, not the state of the code in general."
R.E1 "评论是否针对**意图或变更本身**，而不是针对代码的一般状态。"

R.O1 "Comments that do not pertain to the change as a whole, should refer directly to the code elements that should be modified in order for approval (e.g., variable name, line of code)."
R.O1 "不涉及整个变更的评论，应直接引用**为了获得批准而应修改的代码元素（例如，变量名，代码行）**。"

R.O2 "A relevant comment is one that is specific to the change and shows a deep understanding of the code."
R.O2 "相关的评论是针对变更的具体评论，并**显示出对代码的深刻理解**。"

4.1.2 Informativeness. If the code review comment provides sufficient information.
4.1.2 信息性。代码审查评论是否提供了充分的信息。

Evaluation Criteria. Similar to the process discussed in the evaluation criteria of Relevance, there are 2 essential and 2 optional evaluation criteria corresponding to the attribute of Informativeness.
评估标准。与在相关性评估标准中讨论的过程类似，有 2 个基本和 2 个可选的评估标准对应于信息性属性。

I.E1 Clear intention. The CRC clearly specifies its intention (i.e., what is the further action needed) to make sure the CRC is actionable.
I.E1 意图清晰。CRC 清楚地说明了其意图（即需要采取什么进一步行动），以确保 CRC 具有可操作性。

The intention can include: 1) raising a question and asking for an answer; 2) identifying a problem that should be fixed; 3) providing suggestions that may be non-blocking and not urgent to take action.
意图可以包括：1）提出问题并寻求答案；2）指出应修复的问题；3）提供可能非阻塞且不需要紧急采取行动的建议。

I.E2 Provide reason or context information. Based on the intention, provide context in the CRC.
I.E2 提供理由或背景信息。根据意图，在 CRC 中提供背景信息。

For example, (1) questioning: specifying what is the point of the question (e.g., not just "Why?"); (2) identifying issues: explaining what is the problem; (3) providing suggestions: the reason of such suggestions.
例如，（1）提问：说明问题的重点是什么（例如，不仅仅是"为什么？"）；（2）指出问题：解释问题是什么；（3）提供建议：提出此类建议的理由。

I.O1 Provide suggestions for the next step. Try to provide suggestions for the next step if available.
I.O1 提供下一步建议。如果可能，尽量提供下一步的建议。

I.O2 Provide reference information. The CRC provides reference information that might be helpful to the target developer; such information may include the link to reference documents, guidelines, code, etc.
I.O2 提供参考信息。CRC 提供可能对目标开发人员有帮助的参考信息；此类信息可能包括参考文档、指南、代码等的链接。

Discussion. As shown in Figure 3, over 85% of the participants consider that Informativeness is important to the clarity of CRCs.
讨论。如图 3 所示，超过 85% 的参与者认为信息性对 CRC 的清晰度很重要。

The remaining participants consider its importance as neutral or somewhat unimportant.
其余参与者认为其重要性为中立或比较不重要。

However, the participants do not leave comments regarding the potential reasons.
然而，参与者没有留下关于潜在原因的评论。

Below, we present the comments from our survey participants for their suggested evaluation criteria on Informativeness.
下面，我们展示调查参与者关于他们对信息性建议评估标准的评论。

Their comments are corresponded to the evaluation criteria discussed above and the related part is marked in bold.
他们的评论与上面讨论的评估标准相对应，相关部分以粗体标记。

I.E1 "It should be immediately obvious after reading the comment what the commenter wants me to do, why, and why their version is better than my version of the change."
I.E1 "读完评论后应该立即清楚**评论者想要我做什么**，为什么，以及为什么他们的版本比我的变更版本更好。"

"One of the most important aspects to me is if the intent of the comment is clear."
"对我来说最重要的方面之一是**评论的意图是否清晰**。"

I.E2 “Including reason or context of why the code review comment is made.”
I.E2 “包括提出代码审查评论的**理由或背景**。”

“Whether the comment contains a reasoning for why the change is wrong and needs to be amended.”
“评论是否包含**关于为什么变更错误且需要修正的推理**。”

I.O1 “if the comment is rejecting a change, it should at least include a suggestion of an alternative approach.”
I.O1 “如果评论拒绝某个变更，它至少应该包括**对替代方法的建议**。”

I.O2 “Pointers and references to the materials and existing discussions in the wild are important.”
I.O2 “**指向材料和现有讨论的指针和参考**非常重要。”

4.1.3 Expression. If the code review comment is readable, easy to understand, and friendly.
4.1.3 表达。代码审查评论是否具有可读性、易于理解且友好。

Evaluation Criteria. There are 2 essential and 2 optional evaluation criteria corresponding to the attribute of Expression.
评估标准。有 2 个基本和 2 个可选的评估标准对应于表达属性。

E.E1 Concise and to-the-point. Describe the idea as precise and concise as possible to avoid vagueness, ambiguity, and incoherence.
E.E1 简洁明了。尽可能精确和简洁地描述想法，以避免含糊、歧义和不连贯。

E.E2 Polite and objective. The CRC should express the idea in a polite manner, and focus on the code rather than the person.
E.E2 礼貌客观。CRC 应以礼貌的方式表达想法，并关注代码而不是人。

E.O1 Readable format. The CRC is written in a human readable format.
E.O1 可读格式。CRC 以人类可读的格式编写。

E.O2 Proper syntax and grammar. The CRC is written in a correct syntax and grammar, without typos or incomplete words.
E.O2 正确的句法和文法。CRC 以正确的句法和文法编写，没有拼写错误或不完整的单词。

Discussion. As shown in Figure 3, 78.7% (i.e., 40.8% very important + 37.9% somewhat important) of the survey participants acknowledge the importance of Expression to the CRC’s clarity.
讨论。如图 3 所示，78.7%（即 40.8% 非常重要 + 37.9% 比较重要）的调查参与者承认表达对 CRC 清晰度的重要性。

There are 12.6% of the participants consider its importance as neutral and 8.8% as unimportant.
有 12.6% 的参与者认为其重要性为中立，8.8% 的人认为不重要。

For example, one participant that selects neutral comments “It dependes on what the expression be used for. A long but easy to understant expression is ok”.
例如，一位选择中立的参与者评论道：“这取决于表达用于什么。长但易于理解的表达是可以的”。

One participant that selects somewhat unimportant comments “Really depends on the working relations between the developer and the reviewer. As long as they can understand each-other all is well”.
一位选择比较不重要的参与者评论道：“实际上取决于开发人员和审查者之间的工作关系。只要他们能互相理解就好了”。

Overall, Expression has a relatively lower positive rate compared to the other two attributes, but still accounts for the majority of the participants.
总体而言，与其他两个属性相比，表达的积极率相对较低，但仍占参与者的大多数。

Below, we present the comments regarding the evaluation criteria suggested by our survey participants who acknowledge the importance of Expression.
下面，我们展示承认表达重要性的调查参与者对建议评估标准的评论。

E.E1 “When evaluating the expression of a code review comment, you’re looking at how well the feedback is communicated, whether it is clear, concise, and effectively conveys the reviewer’s thoughts”
E.E1 “在评估代码审查评论的表达时，你要看反馈的传达效果如何，是否清晰、简洁，并有效地传达了审查者的想法”

E.E2 “Comments should have friendly tone and comment on the code, not the person.”
E.E2 “评论应该有友好的语气，并且评论代码，而不是人。”

E.O1 “Formatting around non-english or code snippets. (i.e. backticks “). This helps improve overall clarity.”
E.O1 “围绕非英语或代码片段的格式化。（即反引号 `）。这有助于提高整体清晰度。”

E.O2 “Comment should be plain human readable sentences, because PR author and other reviewers are humans. Proper syntax and grammar, absence of typos are important as well.”
E.O2 “评论应该是普通的人类可读句子，因为 PR 作者和其他审查者都是人类。正确的句法和文法，没有拼写错误也很重要。”

4.1.4 Practitioners’ Feedback on Additional Attributes. Apart from the three attributes, we also ask the participants for additional aspects or attributes that may contribute to the clarity of CRCs.
4.1.4 从业者对额外属性的反馈。除了这三个属性外，我们还询问参与者是否有其他方面或属性可能有助于 CRC 的清晰度。

In total, we receive 48 responses to this question.
我们总共收到了 48 份针对该问题的回复。

After removing three responses that are “N/A” or “None”, we analyze the remaining 45 responses and summarize them as follows.
在删除了三个“不适用”或“无”的回复后，我们分析了剩余的 45 份回复，并总结如下。

Note that a response may be summarized into multiple aspects.
请注意，一个回复可能会被总结为多个方面。

• Consistent to our attributes or evaluation criteria. We find that the comments of 31 participants are consistent to our attributes or the evaluation criteria.
• 与我们的属性或评估标准一致。我们发现 31 位参与者的评论与我们的属性或评估标准一致。

For example, “Providing references (links to other discussions, code changes, documentation, etc.) is important for building trust and minimizing the length of feedback cycles” is consistent to I.O2 and “Politeness” is consistent to E.E2.
例如，“提供参考（其他讨论、代码变更、文档等的链接）对于建立信任和最小化反馈周期长度很重要”与 I.O2 一致，“礼貌”与 E.E2 一致。

• Overall expectations on CRCs. There are 11 participants comment their expectations on CRCs, which may not be directly related to clarity.
• 对 CRC 的总体期望。有 11 位参与者评论了他们对 CRC 的期望，这可能与清晰度没有直接关系。

For example, “The reviewer should be responsive i.e. should quickly respond to the questions raised by the contributor” is about the time taken to reply and “Some reviews will require a history to develop how experienced the reviewee is so the appropriate level of explanation is given for their skill” is about writing CRCs based on reviewee’s knowledge.
例如，“审查者应该反应迅速，即应该快速回应贡献者提出的问题”是关于回复所需的时间，而“有些审查需要历史记录来了解被审查者的经验丰富程度，以便针对他们的技能水平给予适当程度的解释”是关于根据被审查者的知识编写 CRC。

• Other suggestions on code review. There are 8 participants comment on other suggestions regarding the practice of code review.
• 关于代码审查的其他建议。有 8 位参与者评论了关于代码审查实践的其他建议。

For example, “Rich features of the code review tool is also important. For example, GitHub provides a "suggestion" feature, it makes the suggestion of the code change clear” is about leveraging tools to provide suggestions and “I’m not sure if this it too meta, but I think having a space to talk about what kind of culture you want to encourage is important to setting standards” is about the role of code review in building relationship and culture.
例如，“代码审查工具的丰富功能也很重要。例如，GitHub 提供了一个‘建议’功能，它使代码变更的建议变得清晰”是关于利用工具提供建议，而“我不确定这是不是太元了，但我认为有一个空间来讨论你想要鼓励什么样的文化对于设定标准很重要”是关于代码审查在建立关系和文化中的作用。

Overall, there is no additional attribute derived from the practitioners’ feedback on potential new attributes.
总体而言，没有从从业者关于潜在新属性的反馈中得出额外的属性。

However, they provide valuable insights of their expectations on code review, and may inspire future studies to improve the quality of CRCs and the practice of code review.
然而，他们提供了关于代码审查期望的宝贵见解，并可能激发未来的研究，以提高 CRC 的质量和代码审查的实践。

Summary of RQ1:
RQ1 总结：

Based on our literature review and survey with open-source practitioners, we derive three attributes related to the clarity of CRCs and their corresponding evaluation criteria.
基于我们的文献综述和对开源从业者的调查，我们得出了三个与 CRC 清晰度相关的属性及其相应的评估标准。

A majority of the practitioners consider these three attributes important to the clarity of CRCs.
大多数从业者认为这三个属性对 CRC 的清晰度很重要。

4.2 RQ2: Clarity of CRCs in Open-Source Projects
4.2 RQ2：开源项目中 CRC 的清晰度

In this RQ, we first present our detailed process on analyzing the results of our manual investigation.
在本研究问题中，我们首先介绍分析人工调查结果的详细过程。

We then present the results of our quantitative analysis and case study, respectively.
然后，我们分别展示定量分析和案例研究的结果。

4.2.1 Experimental Setup. Two authors of this paper independently label the clarity of CRCs following the process and using the datasets discussed in Section 3.
4.2.1 实验设置。本文的两位作者按照第 3 节中讨论的流程和数据集，独立标记 CRC 的清晰度。

Particularly, for each attribute, the CRC will be marked as positive if it meets all of the essential criteria and at least one of the optional criteria.
特别地，对于每个属性，如果 CRC 满足所有基本标准和至少一个可选标准，则标记为阳性。

Otherwise, it will be marked as negative.
否则，将被标记为阴性。

During the data annotation, each attribute is separately and independently labelled, and the results for one indicator will not affect those of the other two attributes.
在数据标注过程中，每个属性都是单独且独立标记的，一个指标的结果不会影响其他两个属性的结果。

When the labelling is completed, the two authors compare their results and discuss each disagreement until reaching a consensus.
当标记完成后，两位作者比较他们的结果并讨论每一个分歧，直到达成共识。

The value of Cohen’s Kappa [41] in this process is 0.87, which indicates a substantial agreement.
在此过程中，Cohen’s Kappa [41] 值为 0.87，这表明具有很高的一致性。

4.2.2 Quantitative Analysis. We present the results of our quantitative analysis on the clarity of CRCs by different programming languages.
4.2.2 定量分析。我们按不同的编程语言展示对 CRC 清晰度的定量分析结果。

Table 4 shows the percentage of CRCs’ clarity for each programming language.
表 4 显示了每种编程语言的 CRC 清晰度百分比。

Specifically, “Negative” refers to the percentage of CRCs that do not meet the evaluation criteria for each attribute, “All positive” refers to the percentage of CRCs that meet the evaluation criteria for all the three attributes.
具体而言，“阴性”是指不符合每个属性评估标准的 CRC 百分比，“全阳性”是指符合所有三个属性评估标准的 CRC 百分比。

Overall, 71.2% of the CRCs meet the evaluation criteria in all of the three attributes, meaning that a large portion of the CRCs (i.e., 28.8%) is not shown to have a sufficient clarity.
总体而言，71.2% 的 CRC 符合所有三个属性的评估标准，这意味着很大一部分 CRC（即 28.8%）未显示出足够的清晰度。

We discuss the results by comparing among the attributes and the programming languages, respectively.
我们分别通过比较属性之间和编程语言之间的结果来进行讨论。

Comparison among attributes. The distribution of CRCs that are negative for different attributes is 11.4% on average for Relevance, 19.3% on average for Informativeness, and 5.8% on average for Expression, respectively.
属性间的比较。不同属性为阴性的 CRC 分布情况分别为：相关性平均 11.4%，信息性平均 19.3%，表达平均 5.8%。

The results show that a non-negligible portion of CRCs in open-source projects is not written with good clarity, especially for Informativeness and Relevance.
结果表明，开源项目中不可忽视的一部分 CRC 书写清晰度不佳，尤其是在信息性和相关性方面。

Comparison among programming languages. We find that the distribution of clarity varies for different programming languages.
编程语言间的比较。我们发现不同编程语言的清晰度分布各不相同。

For example, over 75% of the CRCs for C and Java are all positive.
例如，C 和 Java 超过 75% 的 CRC 都是全阳性的。

Differently, only 63.6% of the CRCs are all positive for C++, meaning that over 35% of its CRCs have an insufficient clarity.
不同的是，C++ 只有 63.6% 的 CRC 是全阳性的，这意味着超过 35% 的 CRC 清晰度不足。

4.2.3 Case Study. For each attribute, we discuss a negative example (i.e., does not meet the evaluation criteria discussed in the experimental setup of this RQ) and a positive example (i.e., meets the evaluation criteria), respectively.
4.2.3 案例研究。对于每个属性，我们分别讨论一个负面示例（即不符合本 RQ 实验设置中讨论的评估标准）和一个正面示例（即符合评估标准）。

Note that we rename the identifier names and slightly rephrase the CRC to avoid directly retrieving the author of the CRC based on provided samples.
请注意，我们重命名了标识符名称并稍微改写了 CRC，以避免根据提供的样本直接检索到 CRC 的作者。

Relevance. As presented in the examples below, the negative example comments “Same here. and also all others”.
相关性。如下面的示例所示，负面示例评论为“这里也一样。还有其他所有地方”。

The CRC itself hardly contains any information relevant to the code change.
CRC 本身几乎不包含任何与代码变更相关的信息。

It may only be relevant to the information outside this code change.
它可能只与此代码变更之外的信息相关。

Therefore, it is not shown to be relevant to the code change, and it is not self-interpretable.
因此，它未显示出与代码变更相关，并且不可自解释。

Therefore, this CRC does not meet the essential criteria of R.E1.
因此，此 CRC 不符合 R.E1 的基本标准。

In comparison, the positive example raises a question, and the question is shown to be relevant to the code change.
相比之下，正面示例提出了一个问题，并且该问题显示与代码变更相关。

Moreover, it specifies the exact location in the code change where the reviewer has a question.
此外，它指定了审查者有疑问的代码变更的确切位置。

Informativeness. As shown in the negative example below, the CRC mentions “This change is not correct”.
信息性。如下面的负面示例所示，CRC 提到“此更改不正确”。

This CRC may imply the developer to revert this change or fix an issue.
此 CRC 可能暗示开发人员撤销此更改或修复问题。

However, the comment does not provide an explanation of why the change is considered incorrect.
然而，该评论没有提供为何认为更改不正确的解释。

It’s important to offer specific reasons or context information to help the developer understand the issue.
提供具体原因或背景信息以帮助开发人员理解问题非常重要。

Therefore, this example CRC does not meet the essential criteria of I.E2.
因此，此示例 CRC 不符合 I.E2 的基本标准。

In comparison, the positive example explains the issue and further provides a suggestion for the further action to reproduce the problem.
相比之下，正面示例解释了问题，并为重现问题的进一步行动提供了建议。

Expression. The CRC in the negative example asks the question in an impolite manner, which does not meet the essential evaluation criteria of E.E2.
表达。负面示例中的 CRC 以不礼貌的方式提问，这不符合 E.E2 的基本评估标准。

According to many of our survey participants’ feedback, being polite and friendly is very important to efficient communications.
根据我们许多调查参与者的反馈，礼貌和友好对于有效沟通非常重要。

Constructive criticism and polite suggestions for improvement are always preferred than harsh or toxic comments.
建设性的批评和礼貌的改进建议总是比严厉或有毒的评论更受欢迎。

Summary of RQ2: We find that a large portion (i.e., 28.8%) of the CRCs in our study open-source datasets have insufficient clarity.
RQ2 总结：我们发现研究的开源数据集中很大一部分（即 28.8%）CRC 缺乏足够的清晰度。

Among the three attributes, Informativeness has the most noticeable insufficiency.
在这三个属性中，信息性的不足最为明显。

4.3 RQ3: Automatically Evaluating the Clarity of CRCs
4.3 RQ3：自动评估 CRC 的清晰度

In this RQ, we present the results of our evaluation on the clarity of CRCs.
在本 RQ 中，我们展示对 CRC 清晰度评估的结果。

4.3.1 Main Results. Table 5 presents the results of ClearCRC, organized by different RIE attributes, model, and metrics.
4.3.1 主要结果。表 5 展示了 ClearCRC 的结果，按不同的 RIE 属性、模型和指标组织。

In the table, we report the average results of five-fold cross validation.
在表中，我们报告了五折交叉验证的平均结果。

The bold number of each column shows the best performance of the corresponding attribute and metric.
每列的粗体数字显示了相应属性和指标的最佳性能。

Comparison among model sets. Overall, pre-trained language models achieve the best performance among all model sets.
模型组间的比较。总体而言，预训练语言模型在所有模型组中取得了最佳性能。

Specifically, with regard to the set average performance across all evaluation attributes, pre-trained language models have the best performance on all four metrics.
具体而言，关于所有评估属性的组平均性能，预训练语言模型在所有四个指标上都具有最佳性能。

For example, the balanced accuracy of set 2 models is 71.25%, outperforming other sets by a large margin (i.e., 17.7% and 20.8% relative improvements compared to set 1 and set 3 models, respectively).
例如，第 2 组模型的平衡准确率为 71.25%，大幅优于其他组（即相对于第 1 组和第 3 组模型分别提高了 17.7% 和 20.8%）。

Apart from balanced accuracy, there is also a consistent advantage for set 2 models on the other three metrics.
除了平衡准确率之外，第 2 组模型在其他三个指标上也具有一致的优势。

We believe that the strong performance of pre-trained language models is attributed to their prior code knowledge and task-specific fine-tuning.
我们认为，预训练语言模型的出色表现归因于它们先前的代码知识和特定任务的微调。

In contrast, large language models perform poorly due to their inability to acquire sufficient knowledge about the clarity of CRCs.
相比之下，大型语言模型表现不佳，因为它们无法获得关于 CRC 清晰度的足够知识。

Comparison among attributes. We find that the models' performance on different attributes varies a lot.
属性间的比较。我们发现模型在不同属性上的表现差异很大。

The average balanced accuracy of Informativeness is 71.30%, while the number is 56.91% for Expression, which may suggest that Expression is a relatively easier attribute for subject models to understand and recognize, but harder for Informativeness.
信息性的平均平衡准确率为 71.30%，而表达为 56.91%，这可能表明对于目标模型来说，表达是一个相对容易理解和识别的属性，而信息性则较难。

However, we also notice that the recall of Informativeness is much lower than Relevance and Expression (i.e., 78.22% compared to 83.41% and 95.87%).
然而，我们也注意到信息性的召回率远低于相关性和表达（即 78.22% 对比 83.41% 和 95.87%）。

We conjecture that since the dataset of Informativeness has a higher proportion of negative samples, the model achieves a higher balanced accuracy in distinguishing positive and negative samples while also making it more difficult to recall negative samples.
我们推测，由于信息性数据集具有较高比例的负样本，模型在区分正样本和负样本方面实现了较高的平衡准确率，同时也使得召回负样本变得更加困难。

Comparison within the set. We mainly analyze the results in set 2 models because they are the best set among all models.
组内比较。我们主要分析第 2 组模型的结果，因为它们是所有模型中最好的一组。

For average performance among all attributes, CodeBERT is better in terms of balanced accuracy (i.e., 73.04% > 69.46%) but behind CodeReviewer on F-1 score (i.e., 93.07% > 94.61%).
就所有属性的平均性能而言，CodeBERT 在平衡准确率方面表现更好（即 73.04% > 69.46%），但在 F-1 分数上落后于 CodeReviewer（即 93.07% > 94.61%）。

Overall, the two models have comparable strengths.
总的来说，这两种模型具有相当的优势。

Considering that the size of CodeBERT (i.e., 125M) is about half of CodeReviewer's (i.e., 223M), we believe CodeBERT demonstrates good generalizability for automatically evaluating CRC's clarity.
考虑到 CodeBERT 的大小（即 125M）约为 CodeReviewer（即 223M）的一半，我们认为 CodeBERT 在自动评估 CRC 清晰度方面表现出良好的通用性。

Overall, the results show that in terms of precision, recall, and F-1 score, the performance of these models is satisfactory (i.e., an overall average score of more than 85%), but there is still room for improvement for balanced accuracy (i.e., 63.58%), which could be credited to the suboptimal ability to detect negative CRC samples.
总体而言，结果表明，在精确率、召回率和 F-1 分数方面，这些模型的性能是令人满意的（即总平均分超过 85%），但在平衡准确率（即 63.58%）方面仍有改进空间，这可能归因于检测负面 CRC 样本的能力欠佳。

Therefore, we believe that ClearCRC is promising in automatically evaluating the clarity of CRCs and further exploration is still required.
因此，我们认为 ClearCRC 在自动评估 CRC 清晰度方面很有前景，仍需进一步探索。

4.3.2 Generalizability On Other Datasets. To evaluate if ClearCRC could generalize to newer or less-studied projects, we conduct a study on a subset of CodeReviewer-New [25], which includes repositories that the original CodeReviewer dataset does not contain, and adopts various approaches to ensure the data quality.
4.3.2 在其他数据集上的通用性。为了评估 ClearCRC 是否可以推广到较新或研究较少的项目中，我们对 CodeReviewer-New [25] 的子集进行了一项研究，该数据集包含了原始 CodeReviewer 数据集中未包含的仓库，并采用了各种方法来确保数据质量。

We randomly sample 135 examples from CodeReviewer-New (i.e., 15 samples for each of the 9 languages), and follow the same data annotation approaches and experimental settings as the main experiments.
我们从 CodeReviewer-New 中随机抽取 135 个示例（即 9 种语言各 15 个样本），并遵循与主要实验相同的数据标注方法和实验设置。

We use the best checkpoint in the fold 1 cross validation of the main experiments for each model.
我们使用每个模型在主要实验的第 1 折交叉验证中的最佳检查点。

Compared to the results with the original CodeReviewer dataset, there is a slight drop of the performance for both models.
与原始 CodeReviewer 数据集的结果相比，两种模型的性能略有下降。

For example, the balanced accuracy decreases by around 3% (i.e., 71.25% -> 68.14%) and the F-1 score decreases by 5% (i.e., 93.84% -> 88.08%).
例如，平衡准确率下降了约 3%（即 71.25% -> 68.14%），F-1 分数下降了 5%（即 93.84% -> 88.08%）。

Considering the different time and project distributions of the two datasets, we think that the decline is still reasonable and ClearCRC could be generalized to the additional datasets.
考虑到两个数据集的不同时间和项目分布，我们认为下降仍在合理范围内，ClearCRC 可以推广到其他数据集。

Summary of RQ3: ClearCRC assisted with pre-trained language models shows promising results for automatic evaluation of the clarity of CRCs, with a balanced accuracy and F-1 score of 71.25% and 93.84%, respectively.
RQ3 总结：结合预训练语言模型的 ClearCRC 在自动评估 CRC 清晰度方面显示出可喜的结果，平衡准确率和 F-1 分数分别为 71.25% 和 93.84%。

Additionally, it could be generalized to newer or less-studied datasets.
此外，它可以推广到较新或研究较少的数据集。

5 Discussion
5 讨论

5.1 Implications
5.1 启示

5.1.1 Implication 1: Actionable guidelines for evaluating and writing clear CRCs.
5.1.1 启示 1：评估和编写清晰 CRC 的可操作指南。

As shown in our results of RQ2, there is a large portion of the CRCs in open-source systems (i.e., 28.8%) that lack clarity.
正如我们在 RQ2 的结果中所示，开源系统中很大一部分 CRC（即 28.8%）缺乏清晰度。

Due to the lack of well-defined guidelines on writing CRCs, it is challenging for developers to write CRCs that can clearly and sufficiently serve as the medium between developers and reviewers.
由于缺乏关于编写 CRC 的明确指南，开发人员很难编写出能够清晰且充分地作为开发人员和审查者之间媒介的 CRC。

Based on our survey with open-source practitioners, many participants mention that they do not want to see CRCs that are “confusing” and “vague”.
根据我们对开源从业者的调查，许多参与者提到他们不希望看到“令人困惑”和“模糊”的 CRC。

Instead, they expect the CRCs to be “clear”.
相反，他们期望 CRC 是“清晰”的。

However, it is also difficult to determine what is a “clear” CRC.
然而，确定什么是“清晰”的 CRC 也很困难。

In our study, we characterize the clarity of CRCs and derive three attributes with their corresponding evaluation criteria.
在我们的研究中，我们表征了 CRC 的清晰度，并推导出了三个属性及其相应的评估标准。

One of our survey participants mentions “I like the idea of thinking more about comments. I would welcome good guidelines.”.
我们的一位调查参与者提到：“我喜欢更多地思考评论这个想法。我会欢迎好的指南。”

Therefore, our findings can be used as actionable guidelines for evaluating and writing clear CRCs, which in turn improves the efficiency and quality of the code review process.
因此，我们的研究结果可以用作评估和编写清晰 CRC 的可操作指南，从而提高代码审查过程的效率和质量。

5.1.2 Implication 2: Select high-quality data for the automated generation of CRCs.
5.1.2 启示 2：为 CRC 的自动生成选择高质量数据。

There are a series of studies that utilize existing CRC data to train models for the automated generation of CRCs [33, 40, 57].
有一系列研究利用现有的 CRC 数据来训练模型以自动生成 CRC [33, 40, 57]。

However, these studies accept all the CRC data in general, without a curation or selection on the quality of data.
然而，这些研究通常接受所有 CRC 数据，没有对数据质量进行整理或筛选。

Based on such a situation, existing CRC generation techniques may learn from CRC data with insufficient clarity and then generate confusing results.
基于这种情况，现有的 CRC 生成技术可能会从清晰度不足的 CRC 数据中学习，进而生成令人困惑的结果。

In our survey, we also ask the participants for their opinion on automated CRC generation techniques.
在我们的调查中，我们还询问了参与者对自动 CRC 生成技术的看法。

They can rate the importance of the clarity of automatically generated CRCs from 1 to 5, where 1 indicates the lowest importance and 5 indicates the hightest importance.
他们可以将自动生成的 CRC 清晰度的重要性从 1 到 5 进行评级，其中 1 表示重要性最低，5 表示重要性最高。

Figure 6 presents the results of their ratings.
图 6 展示了他们的评级结果。

The average rating is 4.04, and more than 70% of the participants consider its importance as 4 or 5.
平均评分为 4.04，超过 70% 的参与者认为其重要性为 4 或 5。

For example, one participant comments that "If I am going to receive automated comments on my code changes, they need to be clear, accurate, and relevant. Otherwise they are just wasting my time".
例如，一位参与者评论道："如果我要收到关于我的代码更改的自动评论，它们需要清晰、准确且相关。否则它们只是在浪费我的时间"。

As shown in the results of RQ3, ClearCRC achieves a high precision in evaluating the clarity of CRCs on all the three attributes.
如 RQ3 的结果所示，ClearCRC 在评估所有三个属性上的 CRC 清晰度方面实现了高精度。

Therefore, the findings of our study can be used for implementing data filtering and selection mechanisms to help identify CRCs with good clarity, improving the overall effectiveness of the automated CRC generation process.
因此，我们的研究结果可用于实施数据过滤和选择机制，以帮助识别清晰度良好的 CRC，从而提高自动 CRC 生成过程的整体有效性。

5.1.3 Implication 3: Provide a more comprehensive quality evaluation for CRCs and its generation.
5.1.3 启示 3：为 CRC 及其生成提供更全面的质量评估。

As discussed above, developers expect to have CRCs with good quality to foster an effective communication among the team members.
如上所述，开发人员期望拥有高质量的 CRC，以促进团队成员之间的有效沟通。

Moreover, existing research on automated CRC generation generally uses BLEU score [44] to examine the textual similarity between the generated CRC and the reference CRC.
此外，现有的自动 CRC 生成研究通常使用 BLEU 分数 [44] 来检查生成的 CRC 与参考 CRC 之间的文本相似度。

While BLEU score can be used to assess the performance of automated CRC generation techniques, it does not directly address the quality of the CRC itself.
虽然 BLEU 分数可用于评估自动 CRC 生成技术的性能，但它并不直接解决 CRC 本身的质量问题。

In other words, a high Bleu score does not necessarily indicate that the generated CRC is clear and concise.
换句话说，高 BLEU 分数并不一定表明生成的 CRC 清晰简洁。

Therefore, the quality of the CRC itself still remains unclear.
因此，CRC 本身的质量仍然不清楚。

In this paper, we study the quality of CRC by understanding and uncovering the clarity of CRC.
在本文中，我们通过理解和揭示 CRC 的清晰度来研究 CRC 的质量。

To do so, we derive the RIE attributes (i.e., Relevance, Informativeness, and Expressiveness) and their respective evaluation criteria.
为此，我们推导出了 RIE 属性（即相关性、信息性和表达性）及其各自的评估标准。

These attributes and criteria can be leveraged to provide a more comprehensive evaluation for CRCs and their automated generation.
这些属性和标准可以被利用来为 CRC 及其自动生成提供更全面的评估。

By utilizing our findings, we aim to provide a more comprehensive quality evaluation for CRCs and their automated generation.
通过利用我们的发现，我们旨在为 CRC 及其自动生成提供更全面的质量评估。

It can help developers in writing better CRCs, and also contribute to the improvement of automated CRC generation techniques, ultimately leading to more effective communication among software development teams.
它可以帮助开发人员编写更好的 CRC，也有助于改进自动 CRC 生成技术，最终促进软件开发团队之间更有效的沟通。

5.2 RIE Indicators and Existing Metrics
5.2 RIE 指标与现有指标

5.2.1 Comparison with Existing Metrics. Despite the RIE attributes we derive from this paper, the community and research have proposed other metrics to evaluate the quality of CRCs such as Readability, Sentiment [7], and Usefulness [47].
5.2.1 与现有指标的比较。尽管我们在本文中得出了 RIE 属性，但社区和研究界已经提出了其他指标来评估 CRC 的质量，例如可读性、情感 [7] 和有用性 [47]。

Below, we compare these existing metrics with our RIE attributes:
下面，我们将这些现有指标与我们的 RIE 属性进行比较：

• Readability: Readability is seen as an important quality dimension of software comments [21], and it emphasizes the difficulty of reading the text (e.g., the number of difficult words and length of the sentence).
• 可读性：可读性被视为软件评论的重要质量维度 [21]，它强调阅读文本的难度（例如，生僻词的数量和句子的长度）。

It is a subset of our Expression indicator (i.e., E.O1), and Expression includes other aspects like the tone.
它是我们表达指标（即 E.O1）的一个子集，表达还包括语气等其他方面。

Besides, a readable CRC could easily be unclear.
此外，可读的 CRC 可能很容易变得不清晰。

For example, a comment generated by LLM is typically very easy to read, but it may contain little valuable and helpful information for further action.
例如，由 LLM 生成的评论通常非常易于阅读，但它可能包含极少对进一步行动有价值和帮助的信息。

• Sentiment [7]: In code review activities, the contributors frequently express positive, neutral, and negative sentiments, and these sentiments are correlated with the complete time of review [7].
• 情感 [7]：在代码审查活动中，贡献者经常表达积极、中立和消极的情感，这些情感与审查完成时间相关 [7]。

Sentiment is related to the evaluation criteria of Expression (i.e., E.E2 "Polite and objective"), as polite and objective comments are usually positive or neutral.
情感与表达的评估标准相关（即 E.E2"礼貌和客观"），因为礼貌和客观的评论通常是积极或中立的。

However, sentiment alone is not enough to assess the clarity of CRC—purely encouraging and positive comments are not necessarily clear, while neutral CRC could achieve sufficient clarity.
然而，仅凭情感不足以评估 CRC 的清晰度——纯粹鼓励和积极的评论不一定清晰，而中立的 CRC 可以达到足够的清晰度。

• Usefulness [47]: Usefulness of one CRC is typically measured based on the outcome of the corresponding code change (e.g., acceptance rate and time) in former research [47], on the assumption that more and faster code changes are useful because they can lead to better software quality.
• 有用性 [47]：CRC 的有用性在以前的研究中通常是基于相应代码变更的结果（例如，接受率和时间）来衡量的 [47]，其假设是更多和更快的代码变更是有用的，因为它们可以导致更好的软件质量。

However, the outcome of one code change depends on many other factors like the identity of the contributor, code style, and the change scope [48], which adds more indeterministic to this measure.
然而，代码变更的结果取决于许多其他因素，如贡献者的身份、代码风格和变更范围 [48]，这增加了该指标的不确定性。

In contrast to Usefulness, which could be deemed as an outcome-driven metric, the RIE attributes are driven by the process of code review activities, mainly focusing on the clarity of CRCs themselves.
与可被视为结果驱动指标的有用性相比，RIE 属性是由代码审查活动的过程驱动的，主要侧重于 CRC 本身的清晰度。

In this paper, instead of introducing a single new metric, we present a set of well-defined and actionable evaluation criteria for assessing the clarity of CRCs.
在本文中，我们没有引入单一的新指标，而是提出了一套定义明确且可操作的评估标准来评估 CRC 的清晰度。

These criteria are derived from a systematic process and align with the shared expectations of practitioners across academia, industry, and the open-source community.
这些标准源自系统化的过程，并与学术界、工业界和开源社区从业者的共同期望保持一致。

5.2.2 Relation Among RIE attributes. The RIE attributes aim to measure the clarity of CRCs from three distinct dimensions.
5.2.2 RIE 属性之间的关系。RIE 属性旨在从三个不同的维度衡量 CRC 的清晰度。

They are conceptually orthogonal, meaning that each dimension assesses a distinct quality aspect and is relatively independent of the others:
它们在概念上是正交的，意味着每个维度评估不同的质量方面，并且相对独立于其他方面：

• Relevance: Relevance primarily assesses the degree and correctness of the relevance between CRCs and code changes.
• 相关性：相关性主要评估 CRC 与代码变更之间相关性的程度和正确性。

• Informativeness: Informativeness mainly evaluates whether one CRC provides useful information like intention, explanation, suggestion, and reference information.
• 信息性：信息性主要评估 CRC 是否提供有用的信息，如意图、解释、建议和参考信息。

• Expression: Expression measures if the CRC is expressed appropriately from perspectives like readability.
• 表达：表达从可读性等角度衡量 CRC 的表达是否恰当。

Since these dimensions capture different aspects, a CRC may perform well in one dimension but poorly in another.
由于这些维度捕捉了不同的方面，一个 CRC 可能在一个维度上表现良好，但在另一个维度上表现不佳。

For example, a CRC may be highly relevant to code but lack meaningful information, or it may contain rich details but be poorly written and hard to understand.
例如，一个 CRC 可能与代码高度相关，但缺乏有意义的信息，或者它可能包含丰富的细节，但写得很差，难以理解。

Because of this, the three dimensions should be evaluated separately to comprehensively assess the clarity of CRCs.
正因为如此，这三个维度应分别评估，以全面评估 CRC 的清晰度。

Besides, we would like to mention that although the concepts and evaluation criteria are independent, some attributes are indeed correlated and likely co-occur.
此外，我们想提一下，虽然概念和评估标准是独立的，但某些属性确实存在相关性，并且可能同时出现。

For example, if one code reviewer is merely complaining about a specific code change, the comment is typically neither informative nor polite.
例如，如果一个代码审查者只是在抱怨特定的代码变更，那么该评论通常既没有信息量也不礼貌。

6 Threats to Validity
6 有效性威胁

6.1 Internal Validity
6.1 内部有效性

We manually label the clarity of CRCs on open-source datasets.
我们在开源数据集上人工标记 CRC 的清晰度。

To mitigate the subjective bias in this process, two authors of this paper label the data independently.
为了减轻这一过程中的主观偏差，本文的两位作者独立标记数据。

The labellers then discuss each disagreement until a consensus is reached.
然后，标注者讨论每一个分歧，直到达成共识。

Following prior studies [18, 32, 35], we use Cohen's Kappa [41] to measure the agreement of the manual investigation results between the two authors.
遵循先前的研究 [18, 32, 35]，我们使用 Cohen's Kappa [41] 来衡量两位作者之间人工调查结果的一致性。

The Cohen's Kappa value in this process is 0.87, which indicates a substantial agreement.
在此过程中，Cohen's Kappa 值为 0.87，这表明具有很高的一致性。

The randomness in the process of our experiments (e.g., splitting the data, training the models) may affect the results.
我们实验过程中的随机性（例如，分割数据、训练模型）可能会影响结果。

To mitigate such threats, we use a five-fold cross validation to conduct the experiments and report the average number in our discussions.
为了减轻这种威胁，我们使用五折交叉验证来进行实验，并在讨论中报告平均数值。

We derive the evaluation criteria by analyzing the 103 survey responses from participants.
我们通过分析来自参与者的 103 份调查回复得出评估标准。

Engaging more experts from various domains may generate more comprehensive results.
邀请更多来自不同领域的专家可能会产生更全面的结果。

6.2 External Validity
6.2 外部有效性

We conduct our study on the datasets proposed by Li et al. [33].
我们在 Li 等人 [33] 提出的数据集上进行研究。

Using other datasets may generate different results and findings.
使用其他数据集可能会产生不同的结果和发现。

However, the datasets of Li et al. [33] extract code changes and the corresponding CRCs from open-source projects written in nine programming languages, which include a diverse range of repositories.
然而，Li 等人 [33] 的数据集从用九种编程语言编写的开源项目中提取代码变更和相应的 CRC，其中包括各种各样的存储库。

We derive the detailed evaluation criteria based on the survey with practitioners from open-source projects written in nine programming languages.
我们根据对用九种编程语言编写的开源项目的从业者的调查，得出了详细的评估标准。

However, the findings of our study are not specialized for specific programming languages and can be generalizable to various projects.
然而，我们的研究结果并非专门针对特定的编程语言，并且可以推广到各种项目。

Our study is conducted based on open-source data and practitioners.
我们的研究是基于开源数据和从业者进行的。

Future studies may verify the generalizability of our findings on industrial systems and projects.
未来的研究可以在工业系统和项目上验证我们发现的普适性。

7 Conclusion
7 结论

In this paper, we investigate how a code review comment (CRC) can clearly and concisely serve as the medium of communication among developers by conducting a multi-phased, comprehensive study.
在本文中，我们通过开展一项多阶段的综合研究，调查了代码审查评论（CRC）如何清晰简洁地作为开发人员之间的沟通媒介。

We derive our RIE attributes of the clarity of CRCs and the detailed evaluation criteria based on the analysis of our literature review and survey with practitioners.
我们基于对文献综述和从业者调查的分析，得出了 CRC 清晰度的 RIE 属性和详细的评估标准。

We also find that a noticeable portion of the CRCs in open-source projects do not have sufficient clarity.
我们还发现，开源项目中很大一部分 CRC 缺乏足够的清晰度。

We further seek to explore the potential of automatically evaluating the clarity of CRCs by proposing an automated framework, namely ClearCRC.
我们进一步寻求通过提出一个名为 ClearCRC 的自动化框架来探索自动评估 CRC 清晰度的潜力。

Experimental results show that ClearCRC is effective in evaluating the clarity of CRCs based on our RIE attributes and outperforms the baseline approaches by a considerable margin.
实验结果表明，ClearCRC 在基于我们的 RIE 属性评估 CRC 清晰度方面是有效的，并且大幅优于基线方法。

Our findings shed light on characterizing the quality of CRCs and further facilitate the collaboration between developers.
我们的发现阐明了 CRC 质量的特征，并进一步促进了开发人员之间的协作。

Data Availability
数据可用性

Our replication package is available and can be accessed using the link [1].
我们的复制包可用，并可通过链接 [1] 访问。

Acknowledgment
致谢

This work was supported by the National Key R&D Program of China (No. 2024YFB4506400), sponsored by CCF-Huawei Populus Grove Fund, and National Natural Science Foundation of China (No. 62172214).
本工作得到了国家重点研发计划（编号 2024YFB4506400）、CCF-华为胡杨林基金以及国家自然科学基金（编号 62172214）的资助。