The 44th ACM/IEEE International Conference on Software Engineering (ICSE 2022)

Code Reviewer Recommendation in Tencent: Practice, Challenge and Direction

Qiuyuan Chen[1*], Dezhen Kong[1*], Lingfeng Bao[1], Chenxing Sun[2], Xin Xia[1], Shanping Li[1]

[1] Zhejiang University, Hangzhou, China
[2] Tencent, Shenzhen, China

[*] Work done while this author was an intern at Tencent.

Motivation

Divided Organizations

Code Review in a relatively small organization

Inner-source Practice

腾讯云 • • •

A lot of Business at Tencent

>70% codes are inner-sourced

Who should review this code change?

~100k Employees

Java

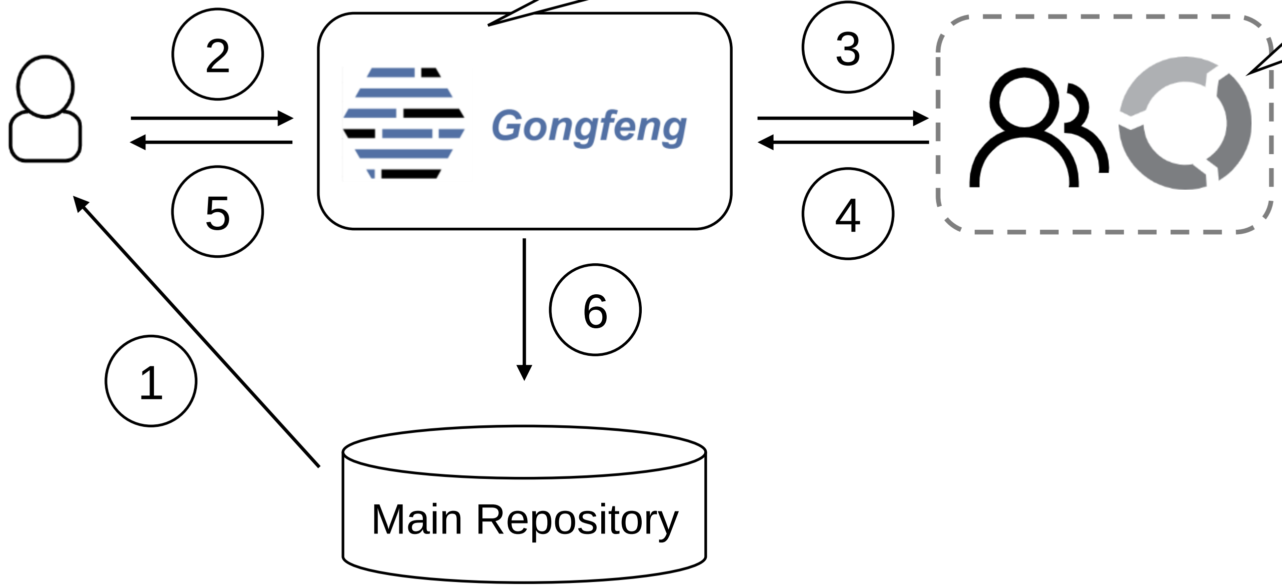
A great many code changes

Virtual organization across groups

Background
Workflow of Modern Code Review

At Tencent, developers use internal code review system “Gongfeng”, like GitHub and Reviewboard.

At Tencent, reviewers are mainly designated by configuration files, or can be robots like *Continuous Integration Tools*.



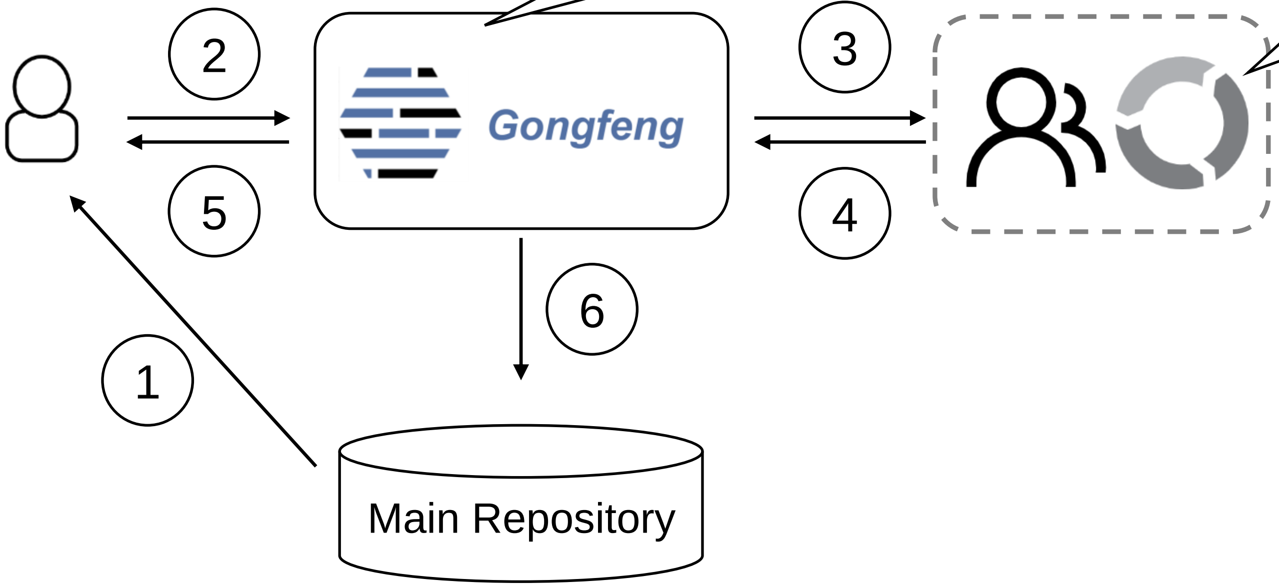
[1] Fetch current code from main repository
[2] Make code changes and push them to CR system
[3] Invite others to review code changes.
[4] Give feedback.
[5] Notify the submitter to improve the code changes
[6] Merge code changes into main repository

Background

Workflow of Modern Code Review

At Tencent, developers use internal code review system “Gongfeng”, like GitHub and Reviewboard.

At Tencent, reviewers are mainly designated by configuration files, or can be robots like *Continuous Integration Tools*.



[1] Fetch current code from main repository
[2] Make code changes and push them to CR system
[3] Invite others to review code changes.
[4] Give feedback.
[5] Notify the submitter to improve the code changes
[6] Merge code changes into main repository

Research Questions

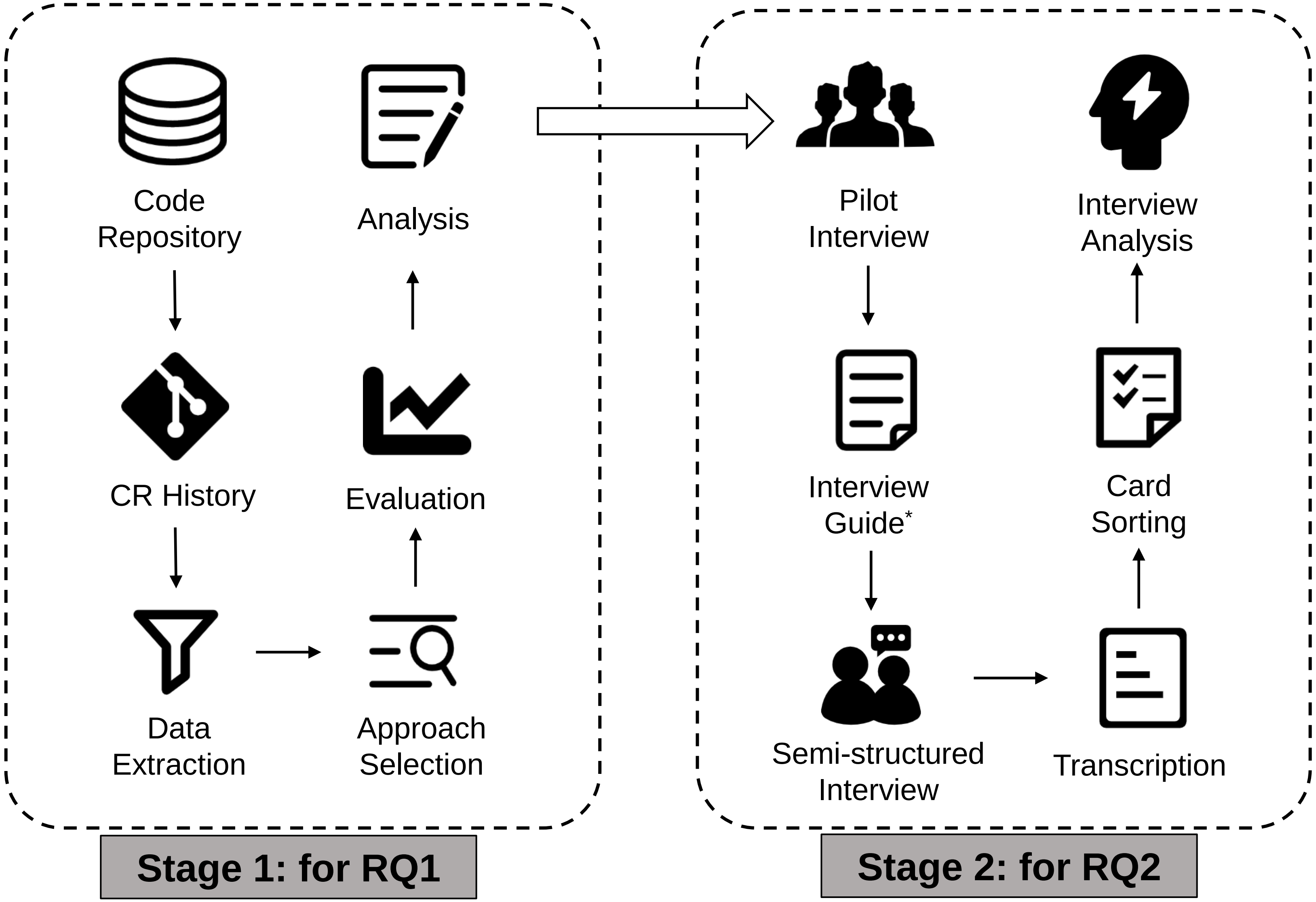
RQ1: What is the effectiveness of code reviewer recommendation approaches on proprietary projects?

We investigate the performance of existing approaches on 10 proprietary projects.

RQ2: What are the perceptions and expectations of practitioners on code reviewer recommendation?

We interview 11 developers to get knowledge about their attitude towards reviewer recommendation systems.

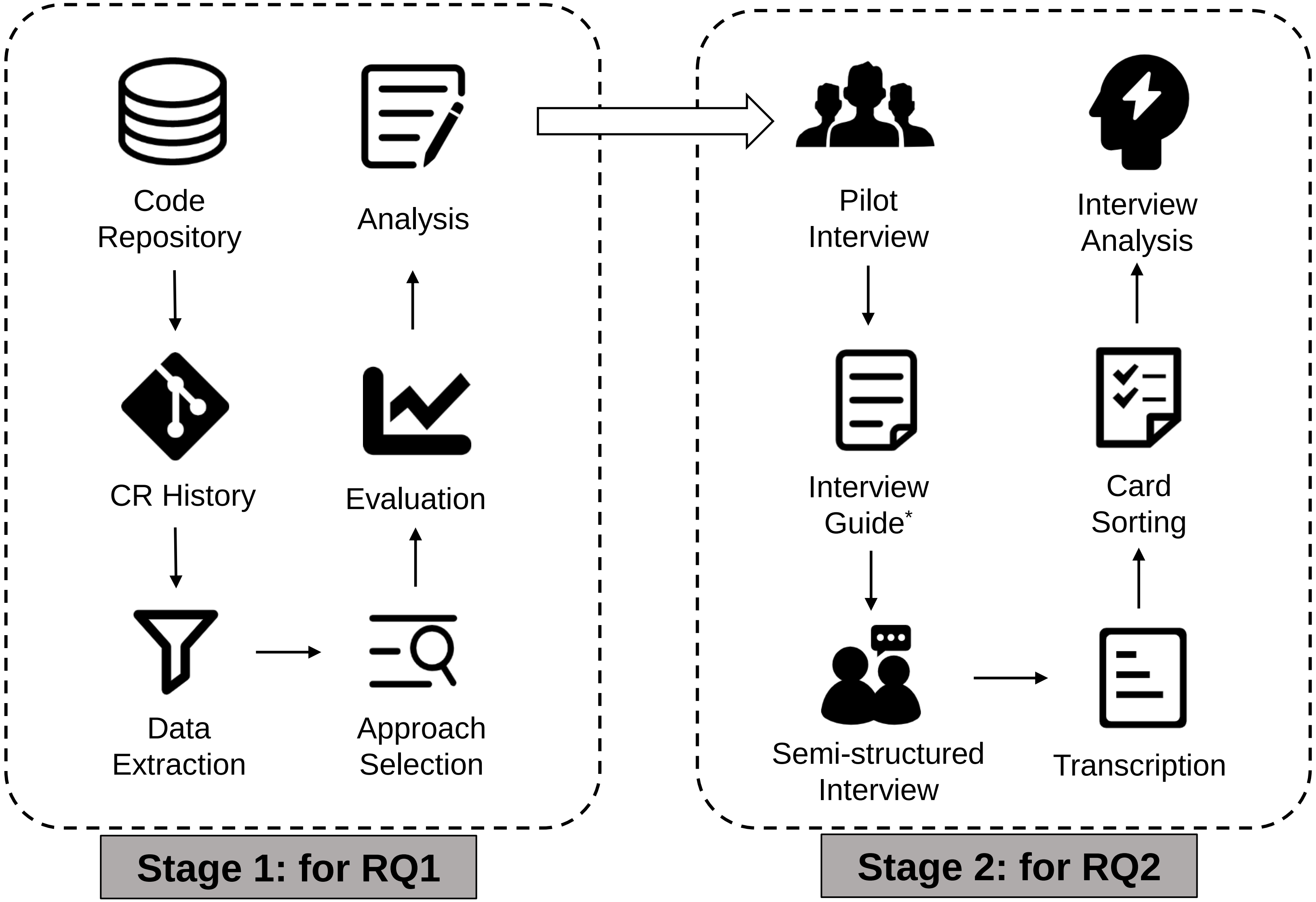
Research Methodology



Interview guide summarization

Part I: Demographic
Part II: Open-ended Discussion
Discussion 2.1: feelings and perceptions
Discussion 2.2: user experience improvements
Part III: Specific Topic Discussion
Discussion 3.1: Existing Practice Feedback
Topic 1: can current CRR system meets need Topic 2: find reviewers in unfamiliar scenario Topic 3: deal with inappropriate reviewers Topic 4: deal with wrongly assigned reviewers Topic 5: Information for selecting reviewers
Discussion 3.2: Code Review Recommendation Scenario
Topic 1: code review scenario Topic 2: inner-source code review experience Topic 3: differences between inner-source and open-source
Discussion 3.3: Code Review Recommendation Algorithm
Topic 1: expected algorithm Topic 2: "hidden information" requests Topic 3: algorithm improvements
Part 4: Statement Agreements

Research Methodology



*Interview guide can be found on https://mfr.osf.io/render?url=https%3A%2F%2Fosf.io%2Fvcqpe%2Fdownload

Research Questions

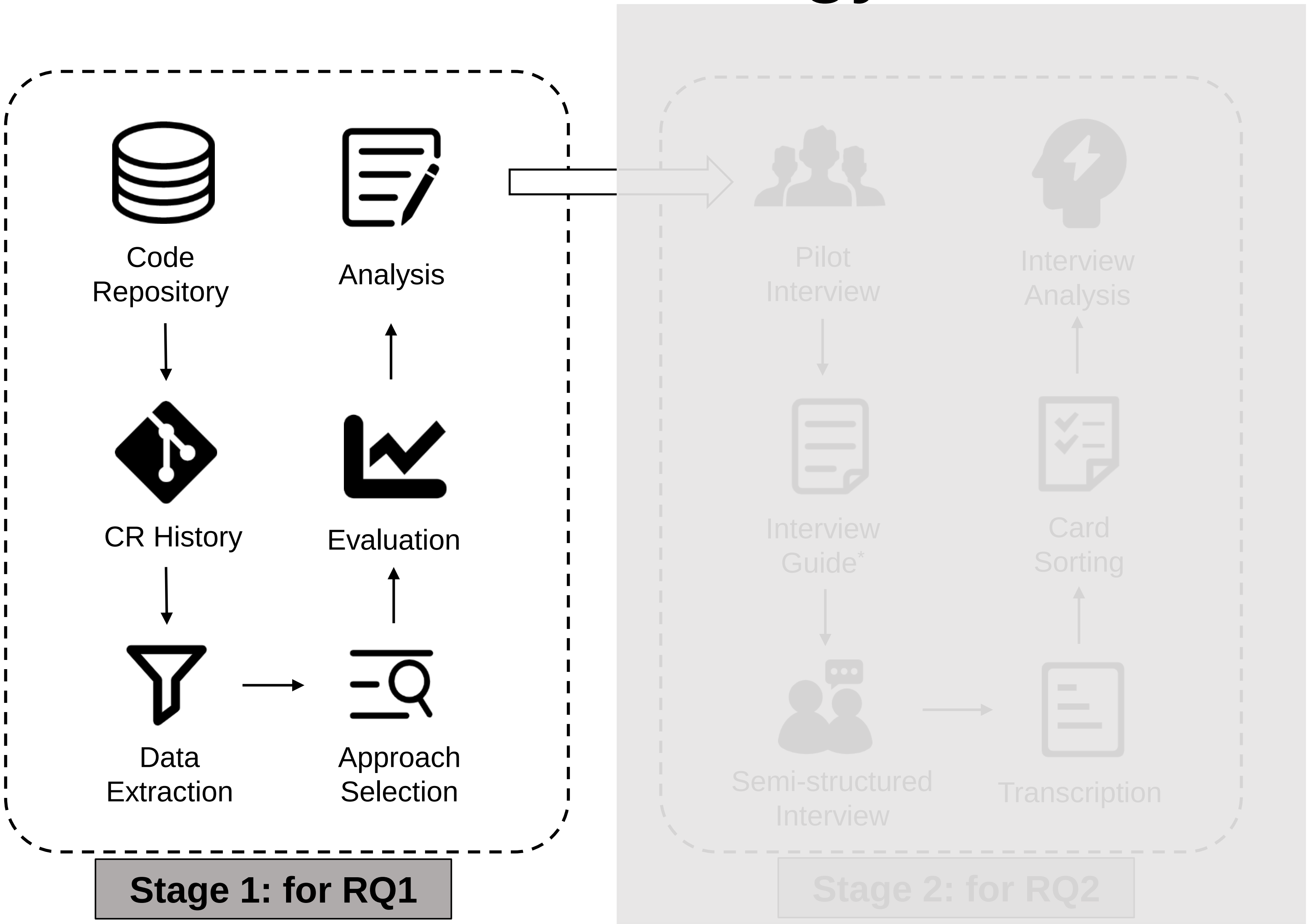
RQ1: What is the effectiveness of code reviewer recommendation approaches on proprietary projects?

We investigate the performance of existing approaches on 10 proprietary projects.

RQ2: What are the perceptions and expectations of practitioners on code reviewer recommendation?

We interview 11 developers to get knowledge about their attitude towards reviewer recommendation systems.

Research Methodology



*Interview guide can be found on https://mfr.osf.io/render?url=https%3A%2F%2Fosf.io%2Fvcqpe%2Fdownload

RQ1 Results

Five Classic Code Reviewer Recommendation Approaches:

- **RevFinder**： is an expertise-based approach that leverages file paths, assuming that the files located in close files may share similar functionality and are likely to be reviewed by reviewers with common experience.
- **TIE**： uses multinomial Naive Bayes to measure the commit message’s textual content (i.e., commit message) similarity and a VSM-based approach to measure the file path similarity.
- **IR (VSM-based)**： vectorizes the PR’s description using VSM, calculates the textual similarities, and ranks the reviewers in the resolved PRs.
- **Comment Network (CN)**： is a recommender that ranks reviewers who share common interests with the contributors of target PR by mining historical comments traces and construct a comment network.
- **cHRev:** considers the reviewing history (review number, review time). It counts the number of comments to the file as part of scores.

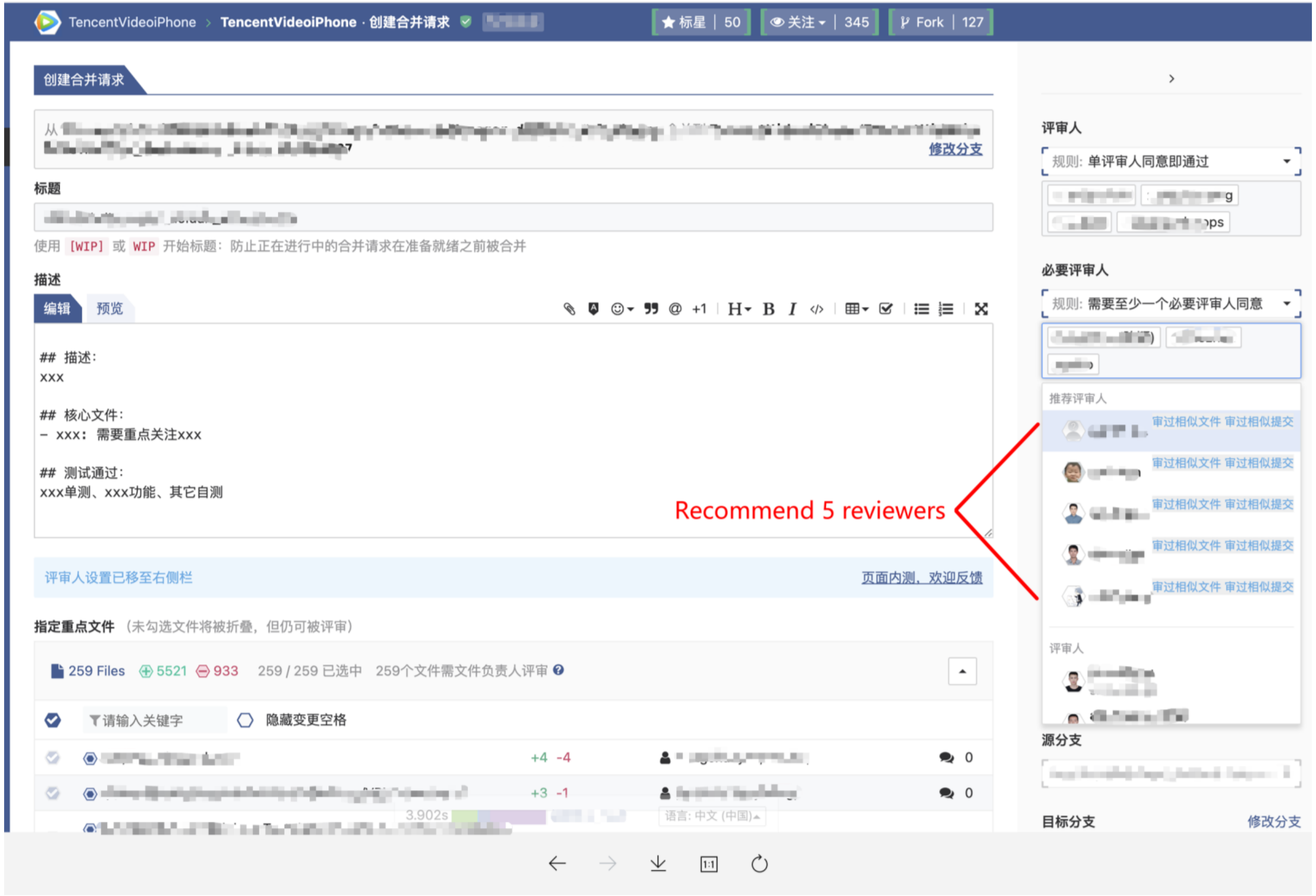
RQ1 Results

Finding 1: Existing approaches do not perform so well on 10 selected projects in Tencent as open-source projects.

Approach	Project	MRR	top1@acc.	top3@acc.	top5@acc.	top10@acc.	top1@prec.	top3@prec.	top5@prec.	top10@prec.	top1@recall	top3@recall	top5@recall	top10@recall
RevFinder	P1	0.16	0.06	0.19	0.30	0.46	0.06	0.06	0.06	0.05	0.06	0.19	0.29	0.45
	P2	0.27	0.14	0.31	0.46	0.60	0.14	0.10	0.09	0.06	0.14	0.31	0.45	0.59
	P3	0.07	0.00	0.17	0.17	0.17	0.00	0.06	0.03	0.02	0.00	0.17	0.17	0.17
	P4	0.17	0.06	0.23	0.31	0.52	0.06	0.08	0.06	0.05	0.06	0.23	0.31	0.52
	P5	0.15	0.13	0.16	0.17	0.18	0.13	0.05	0.03	0.02	0.13	0.16	0.17	0.18
	P6	0.13	0.10	0.16	0.17	0.24	0.10	0.05	0.03	0.02	0.10	0.16	0.17	0.23
	P7	0.20	0.13	0.21	0.29	0.45	0.13	0.07	0.06	0.05	0.10	0.19	0.26	0.41
	P8	0.60	0.33	0.89	0.89	0.93	0.33	0.31	0.20	0.11	0.23	0.72	0.75	0.78
	P9	0.42	0.27	0.51	0.73	0.73	0.27	0.19	0.17	0.09	0.19	0.37	0.59	0.63
	P10	0.50	0.33	0.64	0.73	0.79	0.33	0.24	0.18	0.10	0.21	0.48	0.59	0.67
	Average	**0.27**	**0.16**	**0.35**	**0.42**	**0.51**	**0.16**	**0.12**	**0.09**	**0.06**	**0.12**	**0.30**	**0.38**	**0.46**
TIE	P1	0.37	0.24	0.36	0.53	0.67	0.24	0.12	0.11	0.07	0.21	0.33	0.49	0.63
	P2	0.24	0.11	0.27	0.37	0.57	0.11	0.09	0.07	0.06	0.09	0.21	0.28	0.45
	P3	0.06	0.02	0.04	0.06	0.15	0.02	0.01	0.01	0.01	0.02	0.04	0.06	0.15
	P4	0.16	0.07	0.15	0.22	0.41	0.07	0.05	0.04	0.04	0.07	0.15	0.22	0.41
	P5	0.35	0.20	0.44	0.53	0.60	0.20	0.15	0.11	0.06	0.20	0.44	0.52	0.59
	P6	0.19	0.11	0.23	0.28	0.33	0.11	0.08	0.06	0.03	0.11	0.23	0.28	0.33
	P7	0.21	0.14	0.20	0.26	0.37	0.14	0.07	0.05	0.04	0.12	0.18	0.24	0.35
	P8	0.51	0.28	0.76	0.76	0.80	0.28	0.25	0.16	0.09	0.18	0.47	0.50	0.55
	P9	0.44	0.24	0.55	0.70	0.80	0.24	0.20	0.17	0.10	0.17	0.37	0.52	0.61
	P10	0.46	0.22	0.67	0.74	0.80	0.22	0.24	0.16	0.09	0.17	0.56	0.63	0.69
	Average	**0.30**	**0.16**	**0.37**	**0.45**	**0.55**	**0.16**	**0.13**	**0.09**	**0.06**	**0.13**	**0.30**	**0.37**	**0.48**
IR	P1	0.25	0.07	0.33	0.52	0.71	0.07	0.11	0.10	0.07	0.05	0.28	0.47	0.66
	P2	0.17	0.04	0.18	0.37	0.60	0.04	0.06	0.07	0.06	0.03	0.15	0.29	0.49
	P3	0.02	0.00	0.06	0.06	0.06	0.00	0.02	0.01	0.01	0.00	0.06	0.06	0.06
	P4	0.05	0.00	0.04	0.06	0.30	0.00	0.01	0.01	0.03	0.00	0.04	0.06	0.30
	P5	0.07	0.03	0.09	0.11	0.16	0.03	0.03	0.02	0.02	0.03	0.09	0.11	0.16
	P6	0.08	0.05	0.09	0.13	0.17	0.05	0.03	0.03	0.02	0.05	0.09	0.13	0.17
	P7	0.19	0.13	0.21	0.27	0.39	0.13	0.07	0.05	0.04	0.10	0.18	0.14	0.36
	P8	0.45	0.31	0.44	0.74	0.81	0.31	0.18	0.18	0.11	0.20	0.33	0.54	0.63
	P9	0.20	0.11	0.21	0.31	0.52	0.11	0.08	0.06	0.06	0.07	0.16	0.22	0.37
	P10	0.51	0.36	0.61	0.67	0.80	0.36	0.23	0.16	0.10	0.22	0.44	0.49	0.65
	Average	**0.20**	**0.11**	**0.23**	**0.32**	**0.45**	**0.11**	**0.08**	**0.07**	**0.05**	**0.08**	**0.18**	**0.25**	**0.39**
CN	P1	0.41	0.24	0.51	0.64	0.85	0.24	0.17	0.13	0.09	0.24	0.50	0.63	0.84
	P2	0.67	0.57	0.77	0.83	0.86	0.57	0.26	0.17	0.09	0.56	0.75	0.81	0.85
	P3	0.26	0.20	0.30	0.30	0.50	0.20	0.10	0.06	0.05	0.20	0.30	0.30	0.50
	P4	0.50	0.41	0.57	0.63	0.70	0.41	0.19	0.13	0.07	0.40	0.57	0.63	0.70
	P5	0.58	0.51	0.66	0.70	0.71	0.51	0.22	0.14	0.07	0.50	0.64	0.68	0.70
	P6	0.28	0.21	0.32	0.40	0.47	0.21	0.11	0.08	0.05	0.21	0.32	0.40	0.47
	P7	0.42	0.24	0.57	0.68	0.75	0.24	0.19	0.14	0.07	0.24	0.52	0.63	0.70
	P8	0.60	0.33	0.89	0.93	0.93	0.33	0.32	0.21	0.12	0.23	0.73	0.78	0.83
	P9	0.48	0.33	0.56	0.67	0.80	0.33	0.21	0.17	0.10	0.23	0.42	0.57	0.69
	P10	0.50	0.35	0.62	0.66	0.78	0.35	0.23	0.15	0.10	0.22	0.45	0.49	0.64
	Average	**0.47**	**0.34**	**0.58**	**0.64**	**0.74**	**0.34**	**0.20**	**0.14**	**0.08**	**0.30**	**0.52**	**0.59**	**0.69**
cHRev	P1	0.24	0.16	0.28	0.35	0.47	0.16	0.09	0.07	0.05	0.15	0.27	0.34	0.46
	P2	0.32	0.23	0.35	0.45	0.55	0.23	0.12	0.09	0.06	0.22	0.34	0.44	0.54
	P3	0.04	0.00	0.00	0.00	0.28	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.28
	P4	0.27	0.20	0.31	0.37	0.44	0.20	0.10	0.07	0.04	0.20	0.31	0.37	0.44
	P5	0.49	0.39	0.56	0.63	0.68	0.39	0.19	0.13	0.07	0.38	0.55	0.62	0.67
	P6	0.18	0.12	0.23	0.27	0.30	0.12	0.08	0.05	0.03	0.12	0.23	0.27	0.30
	P7	0.33	0.23	0.41	0.47	0.51	0.23	0.14	0.09	0.05	0.20	0.38	0.43	0.47
	P8	0.11	0.04	0.07	0.07	0.39	0.04	0.02	0.01	0.04	0.02	0.04	0.04	0.27
	P9	0.27	0.17	0.31	0.41	0.49	0.17	0.12	0.10	0.06	0.11	0.24	0.32	0.39
	P10	0.64	0.49	0.74	0.84	0.89	0.49	0.30	0.21	0.12	0.34	0.60	0.71	0.80
	Average	**0.29**	**0.20**	**0.33**	**0.39**	**0.50**	**0.20**	**0.12**	**0.08**	**0.06**	**0.17**	**0.30**	**0.35**	**0.46**

RQ1 Results

Finding 1: Existing approaches do not perform so well on 10 selected projects in Tencent as open-source projects.



RQ1 Results

Finding 1: Existing approaches do not perform so well on 10 selected projects in Tencent as open-source projects.

Approach	Project	MRR	top1@acc.	top3@acc.	top5@acc.	top10@acc.
RevFinder	P1	0.16	0.06	0.19	0.30	0.46
	P2	0.27	0.14	0.31	0.46	0.60
	P3	0.07	0.00	0.17	0.17	0.17
	P4	0.17	0.06	0.23	0.31	0.52
	P5	0.15	0.13	0.16	0.17	0.18
	P6	0.13	0.10	0.16	0.17	0.24
	P7	0.20	0.13	0.21	0.29	0.45
	P8	0.60	0.33	0.89	0.89	0.93
	P9	0.42	0.27	0.51	0.73	0.73
	P10	0.50	0.33	0.64	0.73	0.79
	Average	**0.27**	**0.16**	**0.35**	**0.42**	**0.51**
TIE	P1	0.37	0.24	0.36	0.53	0.67
	P2	0.24	0.11	0.27	0.37	0.57
	P3	0.06	0.02	0.04	0.06	0.15
	P4	0.16	0.07	0.15	0.22	0.41
	P5	0.35	0.20	0.44	0.53	0.60
	P6	0.19	0.11	0.23	0.28	0.33
	P7	0.21	0.14	0.20	0.26	0.37
	P8	0.51	0.28	0.76	0.76	0.80
	P9	0.44	0.24	0.55	0.70	0.80
	P10	0.46	0.22	0.67	0.74	0.80
	Average	**0.30**	**0.16**	**0.37**	**0.45**	**0.55**

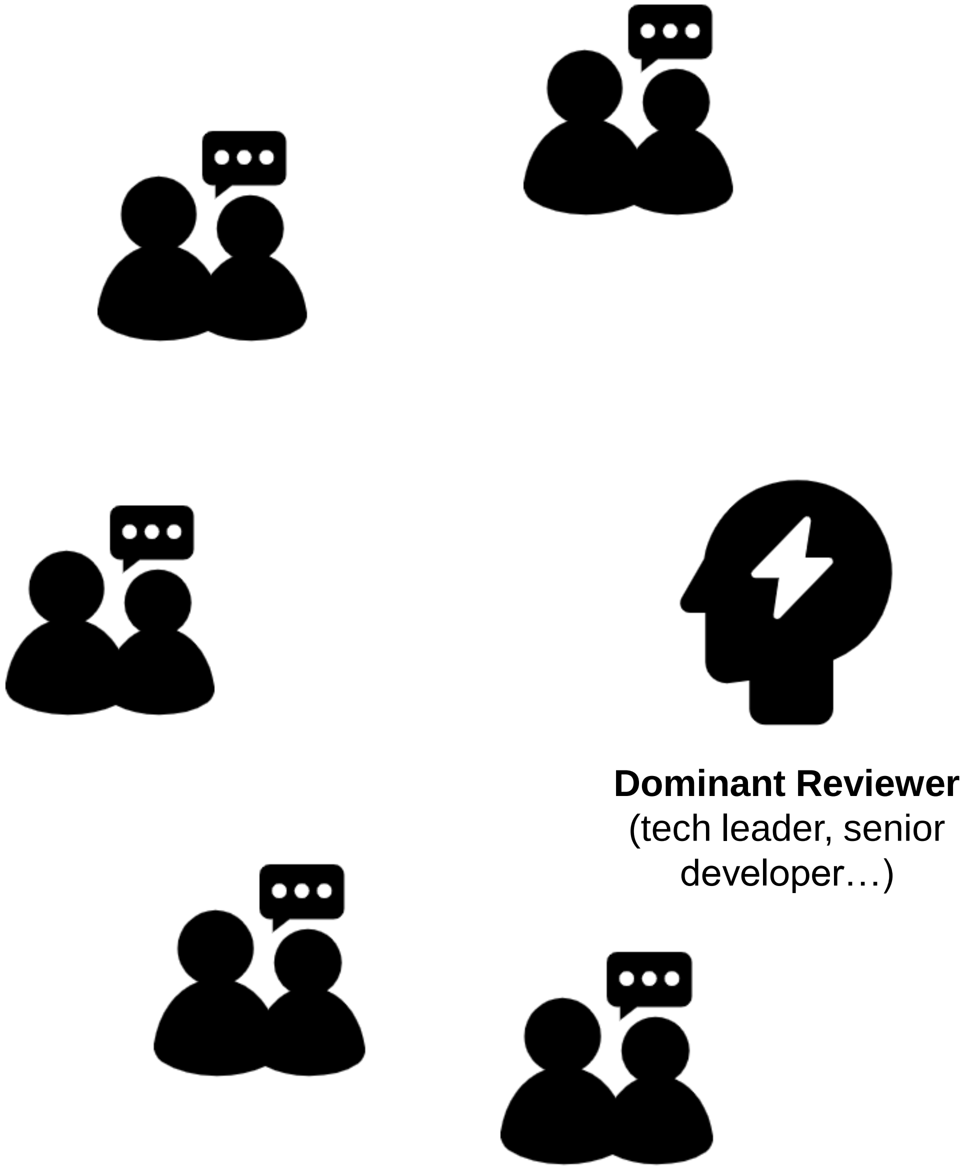
Fig: Performance scores on Proprietary projects

Top-5		
TIE	**Rev.**	**%Imp**
0.87	0.79	**10%**
0.83	0.77	**8%**
0.52	0.41	**27%**
0.93	0.59	**58%**
0.79	**0.64**	**23%**

Fig: Performance scores on Open-source projects

RQ1 Results

- **Finding 2:** Performance of an approach is subject to the characteristics of a project. Projects with dominant reviewers can get good performance.



Beyond the Algorithm: *it is easy to recommend a "correct reviewer", but it is hard to Alleviate the Burden of Dominant Reviewer in practice.*

RQ1 Results

Finding 3: Cold start problem impact the existing approaches.

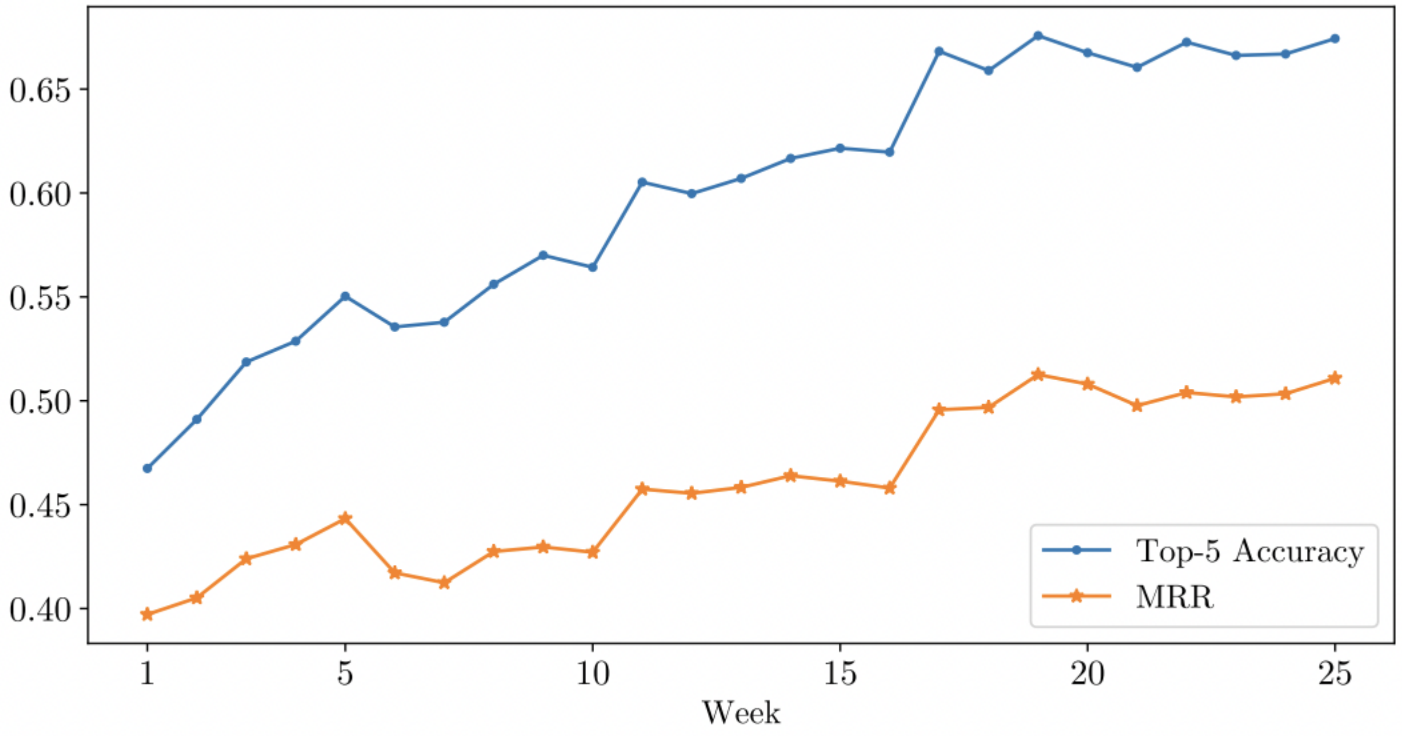


Fig: Average top-5 accuracy and MRR of Comment Network on ten proprietary projects in chronological order.

Code reviewer recommendation approaches suffer from Cold Start Problem and perform badly when initialized.

Research Questions

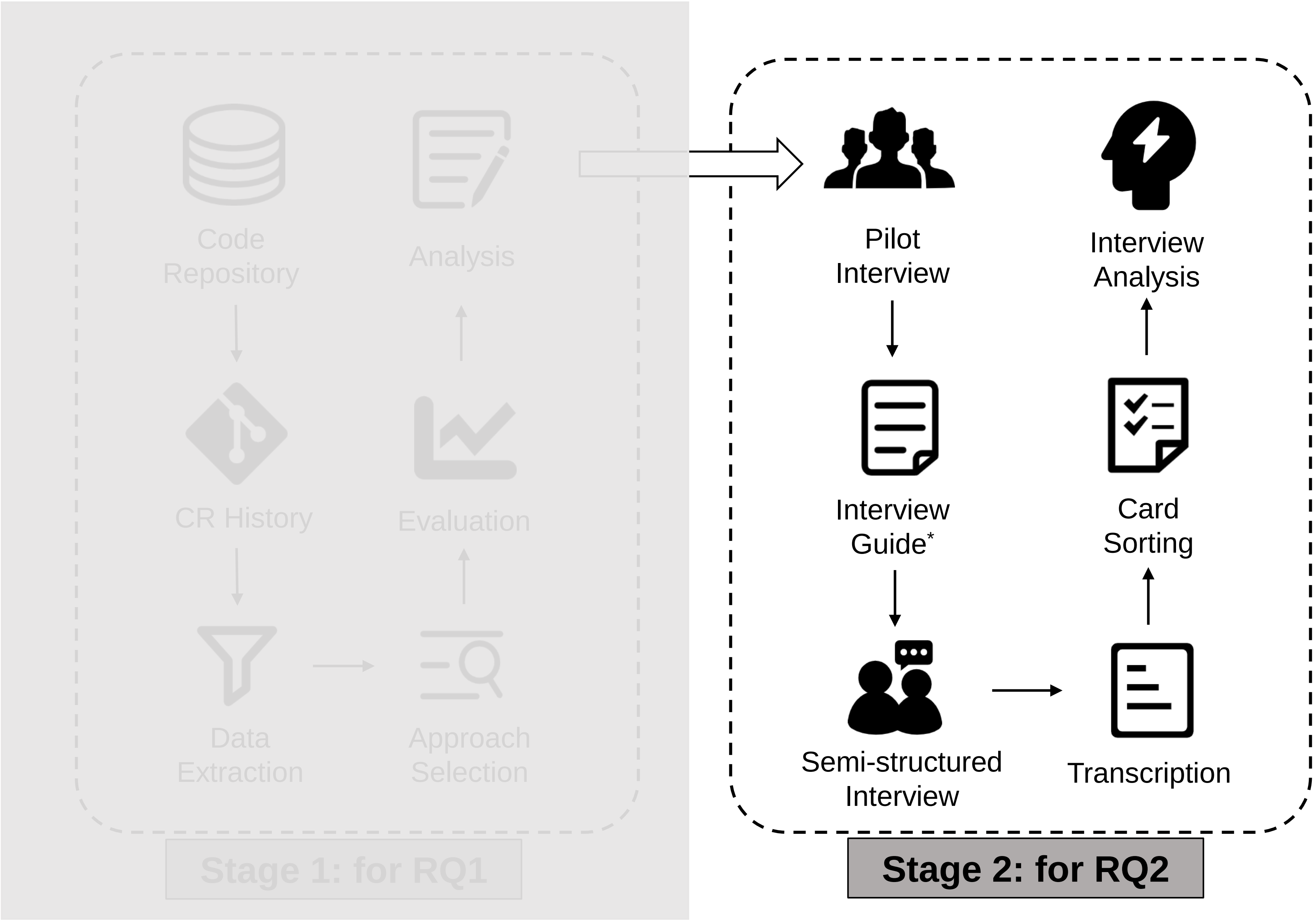
RQ1: What is the effectiveness of code reviewer recommendation approaches on proprietary projects?

We investigate the performance of existing approaches on 10 proprietary projects.

RQ2: What are the perceptions and expectations of practitioners on code reviewer recommendation?

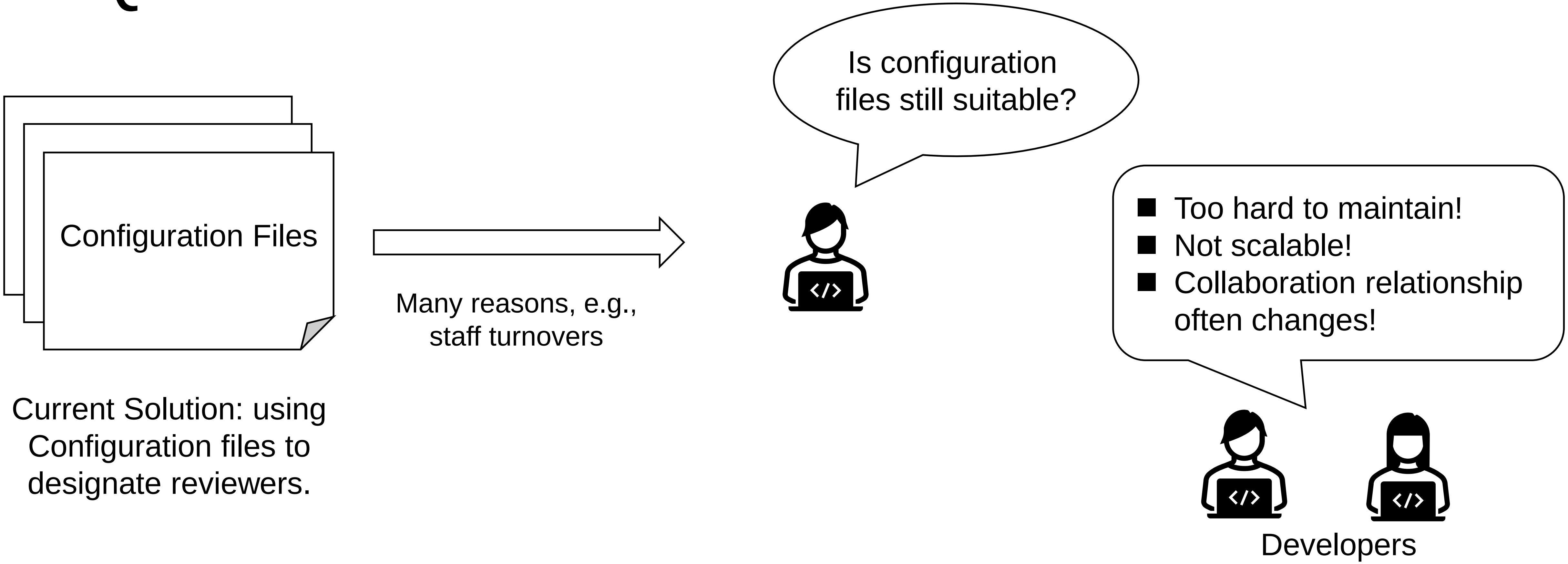
We interview 11 developers to get knowledge about their attitude towards reviewer recommendation systems.

Research Methodology



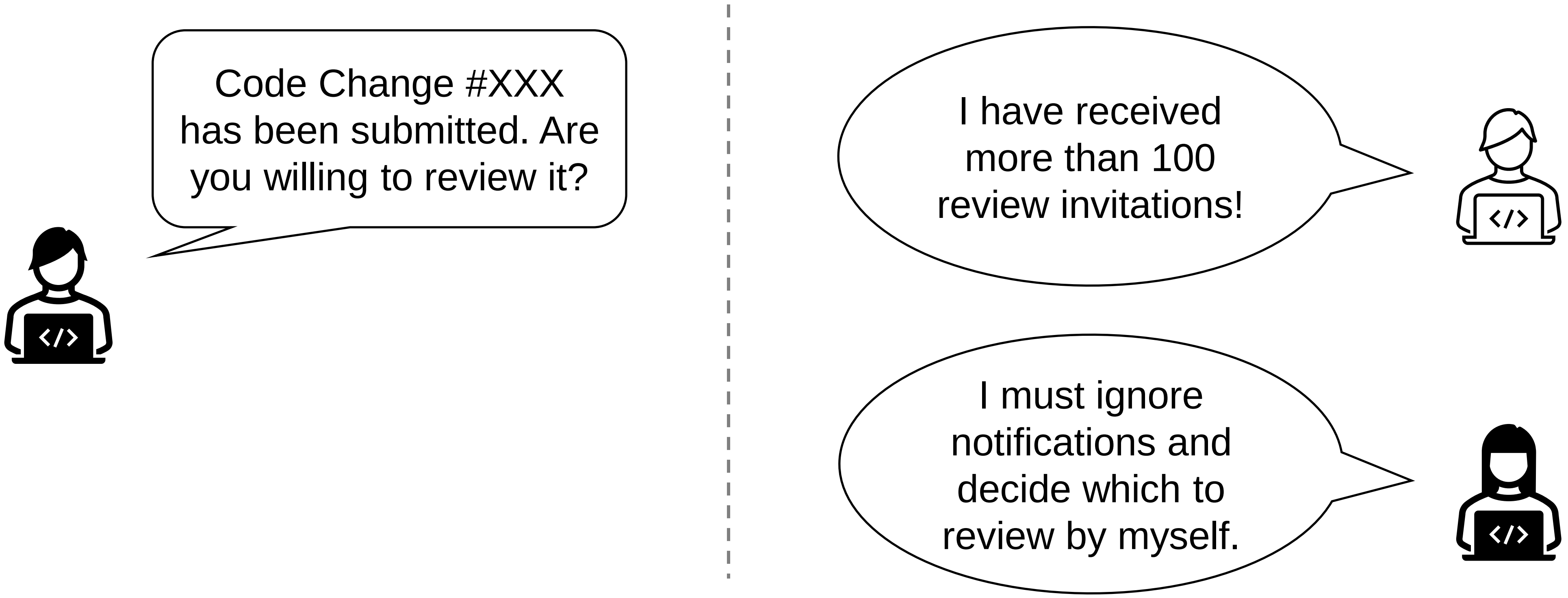
*Interview guide can be found on https://mfr.osf.io/render?url=https%3A%2F%2Fosf.io%2Fvcqpe%2Fdownload

RQ2 Results



Implication: *When the contributor-reviewer relationship is relatively stable, configuration-based recommendations support daily requirements of finding reviewers. However, the manual-maintained configuration cannot assure scalability, and its quality decays quickly.*

RQ2 Results



Implication: *An excessive of invitation in the CRR system can cause "notification noise" for code reviewers, even invalidating the code review invitation process. Code reviewer recommendations should consider the issue and find a tradeoff between the recommendation size and the accuracy.*

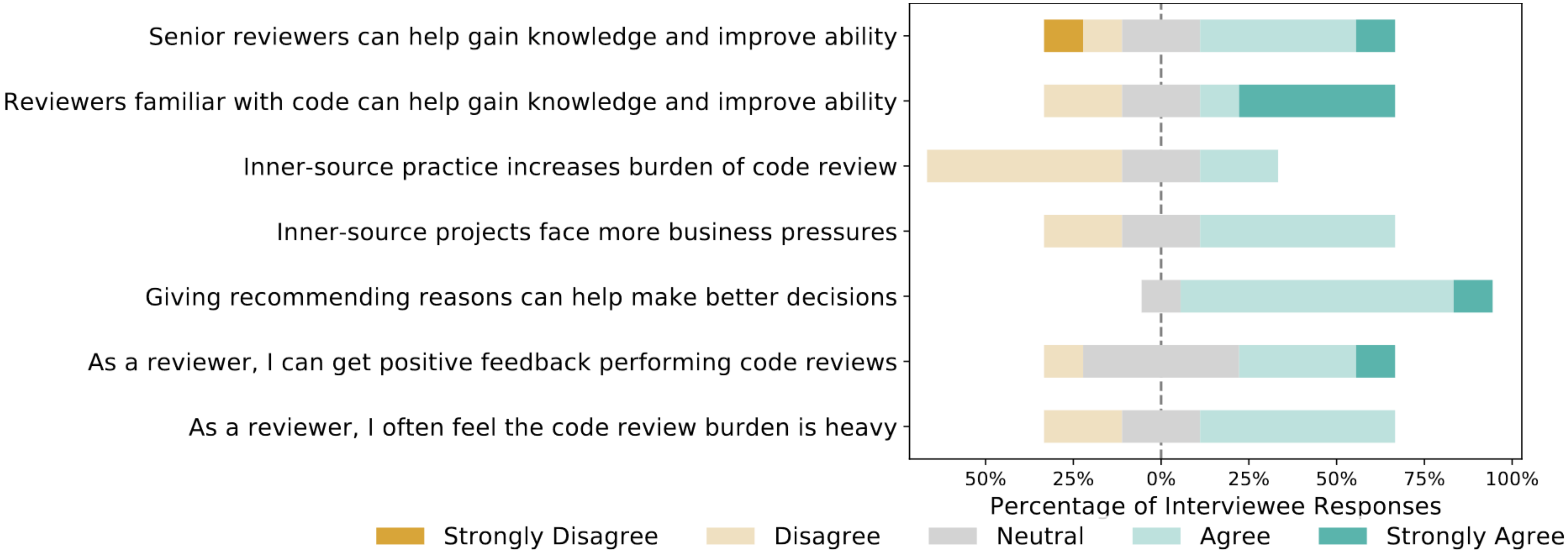
RQ2 Results

In most cases, the historical data-based recommendation approaches are useful.

Sometimes I need a senior partner, rather than recommendation results according to history data.

Implication: *Even though practitioners are confident about the machine-learning-based CRR approaches, a practical CRR system should consider various situations and works in a non-invasive way.*

RQ2 Results



Implication: *Recommendation systems should consider more factors in its working process and bridge the information gap between contributors and reviewers.*

RQ2 Results

Recommendation systems should **consider more factors,** such as learning similar file paths and social network of reviewers.

Recommendation systems should help me know other developers' skills and learn my feedback.

Implication: *Recommendation systems should consider more factors in its working process and bridge the information gap between contributors and reviewers.*

The 44th ACM/IEEE International Conference on Software Engineering (ICSE 2022)

Code Reviewer Recommendation in Tencent: Practice, Challenge and Direction

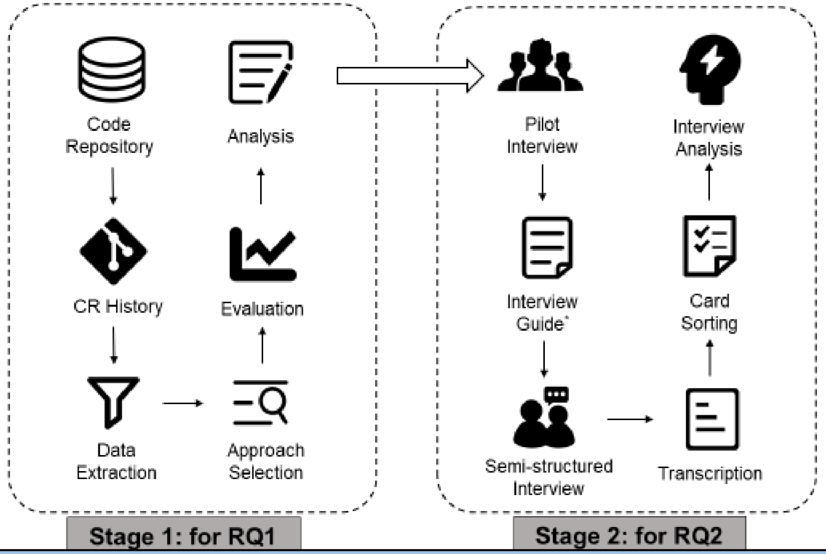
Qiuyuan Chen[1*], Dezhen Kong[1*], Lingfeng Bao[1], Chenxing Sun[2], Xin Xia[1], Shanping Li[1]

[1] Zhejiang University, Hangzhou, China
[2] Tencent, Shenzhen, China
[*] Work done while this author was an intern at Tencent.

Tencent 腾讯

Gongfeng / SAI LAB

Research Methodology



*Interview guide can be found on https://mfr.osf.io/render?url=https%3A%2F%2Fosf.io%2Fvcqpe%2Fdownload

Thank you for listening!

For more details, please refer to our preprint at: www.chenqiuyuan.com.

Contact: chenqiuyuan@zju.edu.cn

RC

Find

selec

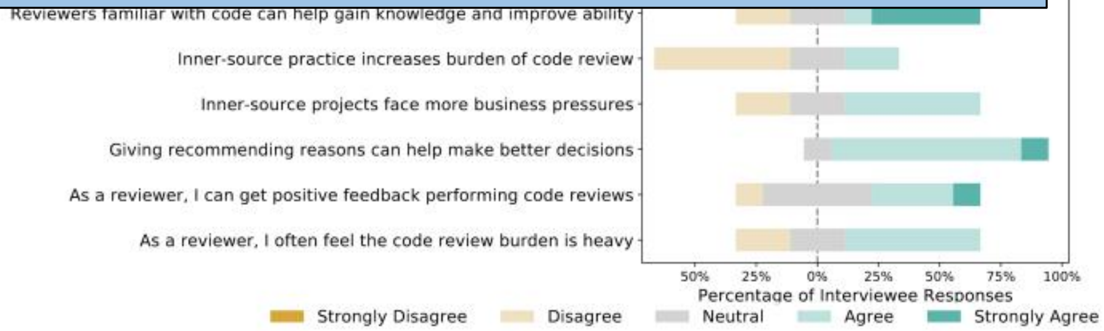
Approach	Project	MRR	top1@acc.	top3@acc.	top5@acc.	top10@acc.
RevFinder	P1	0.16	0.06	0.19	0.30	0.46
	P2	0.27	0.14	0.31	0.46	0.60
	P3	0.07	0.00	0.17	0.17	0.17
	P4	0.17	0.06	0.23	0.31	0.52
	P5	0.15	0.13	0.16	0.17	0.18
	P6	0.13	0.10	0.16	0.17	0.24
	P7	0.20	0.13	0.21	0.29	0.45
	P8	0.60	0.33	0.89	0.89	0.93
	P9	0.42	0.27	0.51	0.73	0.73
	P10	0.50	0.33	0.64	0.73	0.79
	Average	**0.27**	**0.16**	**0.35**	**0.42**	**0.51**
TIE	P1	0.37	0.24	0.36	0.53	0.67
	P2	0.24	0.11	0.27	0.37	0.57
	P3	0.06	0.02	0.04	0.06	0.15
	P4	0.16	0.07	0.15	0.22	0.41
	P5	0.35	0.20	0.44	0.53	0.60
	P6	0.19	0.11	0.23	0.28	0.33
	P7	0.21	0.14	0.20	0.26	0.37
	P8	0.51	0.28	0.76	0.76	0.80
	P9	0.44	0.24	0.55	0.70	0.80
	P10	0.46	0.22	0.67	0.74	0.80
	Average	**0.30**	**0.16**	**0.37**	**0.45**	**0.55**

Fig: Performance scores on Proprietary projects

Top-5		
TIE	Rev.	%Imp
0.87	0.79	10%
0.83	0.77	8%
0.52	0.41	27%
0.93	0.59	58%
0.79	**0.64**	**23%**

Fig: Performance scores on Open-source projects

cent 腾讯



Implication: *Recommendation systems should consider more factors in its working process and bridge the information gap between contributors and reviewers.*