Can Peer Code Reviews be Exploited for Later Information Needs?

Andrew Sutherland

Gina Venolia

Presented by Tianxiao Deng

What this paper did

- Perform an investigation of the practices of software product teams at Microsoft to better understand the nature of the code review(CR) dialog, exchange of information, how the information was retained and the nature of its later use.

Motivation

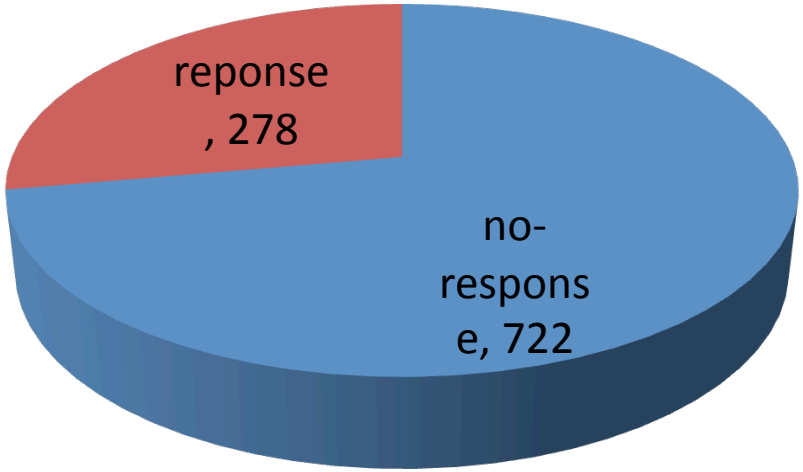
- CR is effective to reduce software defects and improve software quality
- Frequent reviews decrease development time and cost in the long run
- Knowledge exchanged during CR is helpful for engineer to do later modify.

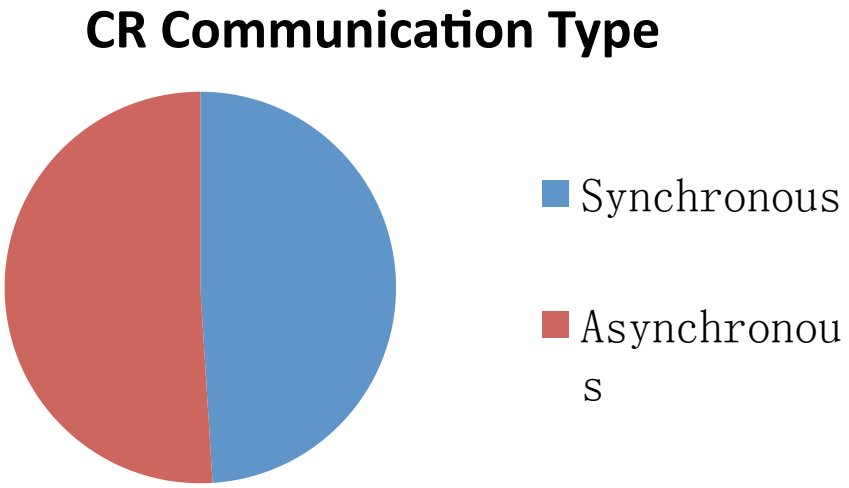
Method

- Design a 25-question survey
- Distribute to random selection of 1000 Microsoft developers and testers using company email
- Participant answer the question using a web form

Result

- Nearly 30% of individuals responded(278 responses)
- 49% of individuals using synchronous communication during CR(e.g talking face to face)
- 51% of individuals using asynchronous communication during CR(e.g sending email)


reponse
, 278
no-
respons
e, 722


CR Communication Type
Synchronous
Asynchronou
s

Result

- Only 18% synchronous CR data was retained

  32% for asynchronous
- Using huge archive of logs to store asynchronous CR (in form of email) but not easy to query
- No easy way to store synchronous CR.

Result

- Developers and testers prefer CR tools with enriched change list and allow to annotate and share change list in a more streamlined fashion
- CR through e-mail usually is task-oriented

  (developers and reviews comment on specific task)

Conclusion

- Code reviews are a enticing opportunity for capturing design rationale,
- Reviews are conducted using a mix of synchronous and asynchronous communication
- Developers think about change lists in a task-oriented way

Conclusion

- Results suggests a tool that

  1. allow task-oriented structuring change lists to be reviewed

  2. allow retaining and querying these structures and discussion based on them

Discussion

- What kind of CR do you prefer? (synchronous or asynchronous)
- Do you have any idea about how to store CR data in synchronous communication?

Reference list

- [1] A.F. Ackerman, P.J. Fowler and R.J. Ebenau, “Software Inspections and the Industrial Production of Software”, Proceedings of a Symposium on Software Validation, Elsevier North Holland, New York, NY, USA, 1984, pp. 13-40.
- [2] A.F. Ackerman, L.S. Buchwalk and F.H. Lewski, “Software Inspections: an Effective Verification P
- [3] L. Brothers, V. Sembugamoorthy, and M. Muller, “ICICLE: Groupware for Code Inspection”, Proceedings of the 1990 ACM Conference on Computer-supported Cooperative Work, ACM, Los Angeles, California, USA, 1990, pp. 169-181. rocess”, IEEE Software, Volume 6(3), May 1989, pp. 31-36.
- [4] M. Fagan, “Design and Code Inspections to Reduce Errors in Program Development”, IBM Systems Journal, Volume 15(3), 1976, pp. 258-287.
- [5] J. Gintell, J. Arnold, J. Houde, J. Kruszelnicki, R. McKenney, and G. Memmi, “Scrutiny: A Collaborative Inspection and Review System”, Proceedings of the 4th European Software Engineering Conference, SpringerVerlag, London, UK, 1993, pp. 344-360.
- [6] J.P. Johnson, “Collaboration, Peer Review, and Open Source Software”, Information Economics and Policy, Volume 18, Elsevier, 2006, pp. 477-497.