

REFERENCES

1. Accessed Jan 2018. *Gerrit*. <https://www.gerritcodereview.com/>.
2. Accessed Jan 2018. *Infer*. <http://fbinfer.com/>.
3. Accessed Jan 2018. *Microsoft Azure*. https://azure.microsoft.com.
4. Accessed Jan 2018. *Phabricator*. <https://www.phacility.com/>.
5. Accessed Jan 2018. *PREfast*. <https://msdn.microsoft.com/en-us/library/ms933794.aspx>.
6. Accessed Jan 2018. *Tricorder*. <https://github.com/google/shipshape>.
7. A. Frank Ackerman, Lynne S. Buchwald, and Frank H. Lewski. 1989. Software Inspections: An Effective Verification Process. *IEEE Softw.* 6, 3 (May 1989), 31–36.
8. Ernesto Arias, Hal Eden, Gerhard Fischer, Andrew Gorman, and Eric Scharff. 2000. Transcending the Individual Human Mind—Creating Shared Understanding Through Collaborative Design. *ACM Trans. Comput.-Hum. Interact.* 7, 1 (March 2000), 84–113.
9. Ernesto Arias, Hal Eden, and Gerhard Fisher. 1997. Enhancing Communication, Facilitating Shared Understanding, and Creating Better Artifacts by Integrating Physical and Computational Media for Design. In *Proceedings of the 2nd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS '97)*. ACM, 1–12.
10. Nathaniel Ayewah, David Hovemeyer, J. David Morgenthaler, John Penix, and William Pugh. 2008. Using Static Analysis to Find Bugs. *IEEE Softw.* 25, 5 (Aug. 2008), 22–29.
11. Alberto Bacchelli and Christian Bird. 2013. Expectations, Outcomes, and Challenges of Modern Code Review. In *Proceedings of the 35th International Conference on Software Engineering (ICSE '13)*. IEEE Computer Society, 712–721.
12. Vipin Balachandran. 2013a. Fix-it: An Extensible Code Auto-Fix Component in Review Bot. In *Proceedings of the 13th IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM '13)*. IEEE Computer Society, 167–172.
13. Vipin Balachandran. 2013b. Reducing Human Effort and Improving Quality in Peer Code Reviews Using Automatic Static Analysis and Reviewer Recommendation. In *Proceedings of the 35th International Conference on Software Engineering (ICSE '13)*. IEEE Computer Society, 931–940.
14. Patti Bao, Elizabeth Gerber, Darren Gergle, and David Hoffman. 2010. Momentum: Getting and Staying on Topic During a Brainstorm. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI '10)*. ACM, 1233–1236.
15. Titus Barik, Kevin Lubick, Samuel Christie, and Emerson R. Murphy-Hill. 2014. How Developers Visualize Compiler Messages: A Foundational Approach to Notification Construction. In *Proceedings of the 2nd IEEE Working Conference on Software Visualization (VISSOFT '14)*. IEEE Computer Society, 87–96.
16. Titus Barik, Justin Smith, Kevin Lubick, Elisabeth Holmes, Jing Feng, Emerson Murphy-Hill, and Chris Parin. 2017. Do Developers Read Compiler Error Messages?. In *Proceedings of the 39th International Conference on Software Engineering (ICSE '17)*. ACM, 575–585.
17. Amiangshu Bosu and Jeffrey C. Carver. 2013. Impact of Peer Code Review on Peer Impression Formation: A Survey. In *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '13)*. IEEE Computer Society, 133–142.
18. Frederick P. Brooks, Jr. 1975. *The Mythical Man-Month* (1st ed.). Addison-Wesley.
19. Cristiano Calcagno and Dino Distefano. 2011. Infer: An Automatic Program Verifier for Memory Safety of C Programs. In *Proceedings of the 3rd International Symposium on NASA Formal Methods (NFM '11 LNCS)*, Vol. 6617. Springer, 459–465.
20. Cristiano Calcagno, Dino Distefano, Jérémy Dubreil, Dominik Gabi, Pieter Hooimeijer, Martino Luca, Peter W. O'Hearn, Irene Papakonstantinou, Jim Purbrick, and Dulma Rodriguez. 2015. Moving Fast with Software Verification. In *Proceedings of the 7th International Symposium on NASA Formal Methods (NFM '15 LNCS)*, Vol. 9058. Springer, 3–11.
21. Maria Christakis and Christian Bird. 2016. What Developers Want and Need from Program Analysis: An Empirical Study. In *Proceedings of the 31st IEEE / ACM International Conference on Automated Software Engineering (ASE '16)*. ACM, 332–343.
22. Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. 2008. Selecting Empirical Methods for Software Engineering Research. In *Guide to Advanced Empirical Software Engineering*. Springer, Chapter 11, 285–311.
23. W. Keith Edwards and Elizabeth D. Mynatt. 1997. Timewarp: Techniques for Autonomous Collaboration. In *Proceedings of the 15th International Conference on Human Factors in Computing Systems (CHI '97)*. ACM, 218–225.
24. K. Anders Ericsson and Herbert A. Simon. May 1980. Verbal Reports as Data. *Psychological review* 87, 3 (May 1980), 215–251.
25. Hamed Esfahani, Jonas Fietz, Qi Ke, Alexei Kolomiets, Erica Lan, Erik Mavrincac, Wolfram Schulte, Newton Sanches, and Srikanth Kandula. 2016. CloudBuild: Microsoft's Distributed and Caching Build Service. In *Proceedings of the 38th International Conference on Software Engineering—Companion Volume*. ACM, 11–20.

26. Michael E. Fagan. 1976. Design and Code Inspections to Reduce Errors in Program Development. *IBM Syst. J.* 15, 3 (Sept. 1976), 182–211.
27. Michael E. Fagan. 1986. Advances in Software Inspections. *IEEE Trans. Softw. Eng.* 12, 1 (Jan. 1986), 744–751.
28. Dror Feitelson, Eitan Frachtenberg, and Kent Beck. 2013. Development and Deployment at Facebook. *IEEE Internet Computing* 17, 4 (July 2013), 8–17.
29. Andre L. Ferreira, Ricardo J. Machado, Jose G. Silva, Rui F. Batista, Lino Costa, and Mark C. Paulk. 2010. An Approach to Improving Software Inspections Performance. In *Proceedings of the 26th IEEE International Conference on Software Maintenance (ICSM '10)*. IEEE Computer Society, 1–8.
30. Nahid Golafshani. 2003. Understanding Reliability and Validity in Qualitative Research. *The Qualitative Report* 8, 4 (Dec. 2003), 597–606.
31. Tovi Grossman and Ravin Balakrishnan. 2008. Collaborative Interaction with Volumetric Displays. In *Proceedings of the 26th International Conference on Human Factors in Computing Systems (CHI '08)*. ACM, 383–392.
32. Reinier M. Hartog. 2015. *Octopull: Integrating Static Analysis with Code Reviews*. Master's thesis. Delft University of Technology.
33. Otmar Hilliges, Lucia Terrenghi, Sebastian Boring, David Kim, Hendrik Richter, and Andreas Butz. 2007. Designing for Collaborative Creative Problem Solving. In *Proceedings of the 6th Conference on Creativity and Cognition (C&C '07)*. ACM, 137–146.
34. Gerard J. Holzmann. 2010. SCRUB: A Tool for Code Reviews. *Innov. Syst. Softw. Eng.* 6, 4 (Dec. 2010), 311–318.
35. Niall Kennedy. 2006. Google Mondrian: Web-based Code Review and Storage. (2006). Accessed Nov 2017.
36. Oleksii Kononenko, Olga Baysal, and Michael W. Godfrey. 2016. Code Review Quality: How Developers See It. In *Proceedings of the 38th International Conference on Software Engineering (ICSE '16)*. ACM, 1028–1038.
37. Oleksii Kononenko, Olga Baysal, Latifa Guerrouj, Yaxin Cao, and Michael W. Godfrey. 2015. Investigating Code Review Quality: Do People and Participation Matter?. In *Proceedings of the 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME '15)*. IEEE Computer Society, 111–120.
38. Thomas D. LaToza, Gina Venolia, and Robert DeLine. 2006. Maintaining Mental Models: A Study of Developer Work Habits. In *Proceedings of the 28th International Conference on Software Engineering (ICSE '06)*. ACM, 492–501.
39. Mark S. Litwin. 1995. *How to Measure Survey Reliability and Validity*. SAGE Publications.
40. Mika V. Mantyla and Casper Lassenius. 2009. What Types of Defects Are Really Discovered in Code Reviews? *IEEE Trans. Softw. Eng.* 35, 3 (May 2009), 430–448.
41. Shane McIntosh, Yasutaka Kamei, Bram Adams, and Ahmed E. Hassan. 2014. The Impact of Code Review Coverage and Code Review Participation on Software Quality: A Case Study of the Qt, VTK, and ITK Projects. In *Proceedings of the 11th Working Conference on Mining Software Repositories (MSR '14)*. ACM, 192–201.
42. Kumiyo Nakakoji, Yasuhiro Yamamoto, Takahiro Suzuki, Shingo Takada, and Mark D. Gross. 1998. From Critiquing to Representational Talkback: Computer Support for Revealing Features in Design. *Knowledge-Based Systems* 11, 7 (Dec. 1998), 457–468.
43. Mehrdad Nurolahzade, Seyed Mehdi Nasehi, Shahedul Huq Khandkar, and Shreya Rawal. 2009. The Role of Patch Review in Software Evolution: An Analysis of the Mozilla Firefox. In *Proceedings of the Joint International and Annual ERCIM Workshops on Principles of Software Evolution and Software Evolution Workshops (IWPSE-Evol '09)*. ACM, 9–18.
44. Teade Punter, Marcus Ciolkowski, Bernd G. Freimut, and Isabel John. 2003. Conducting On-line Surveys in Software Engineering. In *Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE '03)*. IEEE Computer Society, 80–88.
45. Caitlin Sadowski, Jeffrey van Gogh, Ciera Jaspan, Emma Söderberg, and Collin Winter. 2015. Tricorder: Building a Program Analysis Ecosystem. In *Proceedings of the 37th International Conference on Software Engineering (ICSE '15)*. IEEE Computer Society, 598–608.
46. N. Shyamsundar and Rajit Gadh. 2001. Internet-based Collaborative Product Design with Assembly Features and Virtual Design Spaces. *Computer-Aided Design* 33, 9 (Aug. 2001), 637–651.
47. Janet Siegmund, Norbert Siegmund, and Sven Apel. 2015. Views on Internal and External Validity in Empirical Software Engineering. In *Proceedings of the 37th International Conference on Software Engineering (ICSE '15)*. IEEE Computer Society, 9–19.
48. Patanamon Thongtanunam, Shane McIntosh, Ahmed E. Hassan, and Hajimu Iida. 2015. Investigating Code Review Practices in Defective Files: An Empirical Study of the Qt System. In *Proceedings of the 12th Working Conference on Mining Software Repositories (MSR '15)*. IEEE Computer Society, 168–179.
49. Pradeep K. Tyagi. 1989. The Effects of Appeals, Anonymity, and Feedback on Mail Survey Response Patterns from Salespeople. *Journal of the Academy of Marketing Science* 17, 3 (June 1989), 235–241.

50. Alonso H. Vera, Thomas Kvan, Robert L. West, and Simon Lai. 1998. Expertise, Collaboration and Bandwidth. In *Proceedings of the 16th International Conference on Human Factors in Computing Systems (CHI '98)*. ACM, 503–510.
51. Wikipedia. 2017. Gerrit (software). (2017). Accessed Feb 2017.
52. Yunwen Ye, Yasuhiro Yamamoto, and Kumiyo Nakakoji. 2007. A Socio-technical Framework for Supporting Programmers. In *Proceedings of the the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering (ESEC-FSE '07)*. ACM, 351–360.