



Practical remote pair programming : best practices, tips, and techniques for collaborating productively with distributed development teams /

Bolboaca, Adrian,
author

Electronic books

Monografía

Practical Remote Pair Programming provides a hands-on approach to using remote pair programming in your day-to-day work. With this book, you'll learn how to leverage the tools, techniques, and tips and tricks to boost your productivity and enhance your pair programming experience while working in distributed teams

<https://rebiunoda.pro.baratznet.cloud:28443/OpacDiscovery/public/catalog/detail/b2FpOmNlbGVicmF0aW9uOmVzLmJhcmF0ei5yZW4vMjk2MTQ3MzQ>

Título: Practical remote pair programming best practices,tips, and techniques for collaborating productively with distributed development teams Adrian Bolboaca

Editorial: London, England Packt Publishing, Limited [2021] 2021

Descripción física: 1 online resource (240 pages) illustrations

Nota general: Includes index

Contenido: Cover -- Title Page -- Copyright and Credits -- Dedication -- Contributors -- Table of Contents -- Preface -- Section 1: Introduction to Pair Programming -- Chapter 1: Pair Programming and Its Necessity -- The history of pair programming -- Elucidating problems in pairs -- Managing complexity -- Sharpening our knowledge -- Collaborative work -- Leveling knowledge -- Advancing knowledge -- Gaining wisdom -- Improving the system -- Staff liquidity -- Managing complexities in complex domains -- Managing complex domains -- Difficult tasks -- The fastest feedback code review -- Minimizing the defect rate -- Comfort for the future you -- Code is more often read than written -- Exploratory testing with pair programming -- Programming with your CEO -- Social programming -- The rubber duckling effect -- How does pair programming work? -- Knowledge work and knowledge workers -- Time well spent -- Summary -- Further reading -- Chapter 2: How Can Pair Programming Help? -- Defining pair programming -- Driver -- Navigator -- Trainer/facilitator -- Boosting efficiency -- Reducing task switching -- Taking breaks -- Situations when pair programming can help -- Bettering efficiency -- Improving

technical skills -- Aiding knowledge transfer -- Improving communication -- Enhancing problem-solving capabilities -- Simplifying the existing code base -- Situations when pair programming is difficult -- Working alone -- Lack of safe space -- Only I want to pair from my team -- Working better with colleagues with other specializations -- Pairing with a tester -- Pairing with a UI designer -- Pairing with DevOps -- Pairing with a business analyst -- Pair programming in practice -- Learning new things or tricks -- Being social - social programming -- Pair programming cannot solve everything -- Unclear requirements -- Bad coding practices -- Tension within the team Tension within the organization -- Close to deployment time -- Summary -- Further reading -- Chapter 3: Exploring Pair Programming Techniques and Styles -- Understanding pair programming techniques -- The Driver-Navigator technique -- The Pairing-Trainee technique -- The Beginner-Advanced technique -- The Beginner-Beginner technique -- The Ping-Pong technique -- Improving pair programming with styles -- Unplanned pairing -- Traditional pairing -- Elastic pairing -- Strong-style pairing -- Organizing pair programming -- Round-robin pairing -- Promiscuous pairing -- Selective pairing -- How often do we need to pair? -- Should you pair for the whole day? -- Should you pair daily? -- Should you pair weekly? -- Exploring different communication methods -- Aggressive communication -- Submissive communication -- Assertive communication -- Making a difference with the right words, tone, and clarity -- Pair programming best practices -- Taking notes while pair programming -- Starting with some small talk -- Emptying your cup -- Debriefing -- Dialogue courtesy -- Building confidence - committing often and having good unit tests -- Trusting your pair -- Pair programming anti-patterns -- Managing distractions -- Centering the monitor -- Dealing with the "I know it all" attitude -- Addressing small pickings -- Boosting productivity with remote pair programming -- Summary -- Further reading -- Section 2: Remote Pair Programming -- Chapter 4: Using Pair Programming in a Distributed System -- Technical requirements -- Organizing remote pair programming -- The purpose of remote pairing -- Distributed team -- Deciding on the scope -- Duration -- Pomodoro technique -- Schedule -- Kickoff -- Concerns -- Good practices of pairing -- Anti-patterns -- Performing regular retrospectives -- How often we should retrospect -- Retrospective techniques What happens after a retrospective? -- Improving the retrospectives continuously -- Analyzing the results -- Personal introspection -- Tools analysis -- Summary -- Further reading -- Chapter 5: Remote Pair Programming Setup -- Technical requirements -- Checking the internet connection -- Using a cable connection -- Using a Wi-Fi connection -- Using a portable router -- Setting up video and audio -- Setting up video -- Checking the lighting -- Choosing the camera -- Understanding the camera's position -- Checking the audio -- Introducing audio -- Choosing a microphone -- Positioning your microphone -- Use headphones, not loudspeakers -- Mute pairing -- Setting up the IDE -- Key editor functionalities -- Best IDEs for remote pair programming -- IntelliJ IDEA -- Best editor plugins for remote pair programming -- Setting up screen sharing -- Introducing TeamViewer -- Introducing AnyDesk -- Introducing Screen -- Introducing Use Together -- Introducing Tuple -- Introducing Zoom -- Introducing Google Meet -- Introducing Skype -- Learning to use source control -- Source control tools -- Commit often -- Ensemble commits -- Rotation -- Using two computers for coding and remote screening -- Summary -- Further reading -- Chapter 6: Remote Pair Programming-Specific Techniques and Styles -- Recap of main concepts of pair programming -- Understanding general setup -- Understanding the remote driver-navigator technique -- Remote setup -- Remote specifics - driver -- Remote specifics - navigator -- Remote ping-pong technique -- Remote setup -- Remote specifics - driver -- Remote specifics - navigator -- Remote beginner-advanced technique -- Remote setup -- Remote specifics - driver -- Remote specifics - navigator -- Traditional pairing style -- Remote setup -- Remote specifics -- Remote elastic pair programming style -- Remote setup -- Remote specifics Remote strong style -- Remote setup -- Remote specifics -- Good remote practices -- Remote breaks -- Secondary communication channel -- Remote commit approach -- Summary -- Section 3: Tools to Enhance Remote Pair Programming -- Chapter 7: Video and Audio -- Recap of what we have learned so far -- General technical aspects -- Audio is more important than video -- Video settings -- Screen sharing settings -- Quality sound and video for the win -- Learning how to enhance video -- Looking into the camera -- Using a green screen -- Using a virtual background -- Learning to enhance audio -- Speaking into the microphone -- Using an audio compressor -- Performing soundcheck -- Monitoring the sound -- Adding a pop filter -- Enhancing speech -- Employing diction -- Choosing words -- Warming up your voice -- Summary -- Chapter 8: Source Control Rules -- Recap of the source control rules -- Using source control -- Improving source control usage -- Using the commit types -- Understanding commit heuristics -- Committing when part of a feature is done -- Committing when all the tests are written, and green, for a user scenario -- Committing before taking a break, in a stable state -- Committing when the preparatory refactoring is done -- Committing when one characterization test is done -- Committing when one unit test is green -- Summary -- Further reading -- Chapter 9: Remote Access -- Recapping the rules of remote

pairing -- Understanding how remote access tools work -- TeamViewer -- AnyDesk -- Screen -- Chrome Remote Desktop -- Relying on tools -- Bug magnet -- Security for remote access -- Summary -- Further reading -- Why subscribe? -- About Packt -- Other Books You May Enjoy -- Index

ISBN: 1-80056-553-4

Materia: Agile software development Software engineering

Enlace a formato físico adicional: 1-80056-136-9

Baratz Innovación Documental

- Gran Vía, 59 28013 Madrid
- (+34) 91 456 03 60
- informa@baratz.es