Bibliography

[1] Adzic, Gojko, *The Poka-Yoke principle and how to write better software*, blog post at https://gojko.net/2007/05/09/the-poka-yoke-principle-and-how-to-write-better-software, 2007.

[2] Allamaraju, Subbu, *RESTful Web Services Cookbook*, O'Reilly, published 2010.

[3] Atwood, Jeff, *New Programming Jargon*, blog post at https://blog.codinghorror.com/new-programming-jargon, 2012.

[4] Barr, Adam, *The Problem with Software. Why Smart Engineers Write Bad Code*, MIT Press, 2018.

[5] Beck, Kent, and Cynthia Andres, *Extreme Programming Explained: Embrace Change*, Addison-Wesley, published 2004.

[6] Beck, Kent, tweet at https://twitter.com/KentBeck/status/250733358307500032, 2012.

[7] Beck, Kent, *Implementation Patterns*, Addison-Wesley, published 2007.

[8] Beck, Kent, *Naming From the Outside In*, Facebook note at https://www.facebook.com/notes/kent-beck/naming-from-the-outside-in/464270190272517 (accessible without a Facebook account), 2012.

[9] Beck, Kent, *Test-Driven Development By Example*, Addison-Wesley, published 2002.

[10] Beck, Kent, tweet at https://twitter.com/KentBeck/status/1354418068869398538, 2021.

[11] Bernhardt, Gary, *Functional Core, Imperative Shell*, online presentation at https://www.destroyallsoftware.com/screencasts/catalog/functional-core-imperative-shell, 2012.

[12] Böckeler, Birgitta, and Nina Siessegger, *On Pair Programming*, blog post at https://martinfowler.com/articles/on-pair-programming.html, 2020.

[13] Bossavit, Laurent, *The Leprechauns of Software Engineering*, Laurent Bossavit, 2015.

[14] Brooks, Frederick P., Jr., *No Silver Bullet – Essence and Accident in Software Engineering*, 1986. This essay can be found in various sources, and is easily located on the internet. In writing this book, I referred to my copy of *The Mythical Man-Month: Essays on Software Engineering. Anniversary Edition*, Addison-Wesley, published 1995, in which the essay constitutes chapter 16.

[15] Brown, William J., Raphael C. Malveau, Hays W. "Skip" McCormick III, and Thomas J. Mowbray, *AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis*, Wiley Computer Publishing, 1998.

[16] Cain, Susan, *Quiet: The Power of Introverts in a World That Can't Stop Talking*, Crown, 2012.

[17] Campidoglio, Enrico, tweet at https://twitter.com/ecampidoglio/status/1194597766128963584, 2019.

[18] Cirillo, Francesco, *The Pomodoro Technique: The Life-Changing Time-Management System*, Virgin Books, 2018.

[19] Cockburn, Alistair, *Hexagonal architecture*, online article at https://alistair.cockburn.us/hexagonal-architecture/, 2005.

[20] Cohen, Jason, *Modern Code Review* in [75], 2010.

[21] Conway, Melvin E., *How Do Committees Invent?*, Datamation, 1968. I admit that I don't own a copy of the April 1968 issue of Datamation magazine.

Instead, I've used the online reprint that Melvin Conway hosts at http://www.melconway.com/Home/Committees_Paper.html.

[22] Cunningham, Ward, and Bill Venners, *The Simplest Thing that Could Possibly Work. A Conversation with Ward Cunningham, Part V*, interview at www.artima.com/intv/simplest.html, 2004.

[23] Cwalina, Krzysztof, and Brad Abrams, *Framework Design Guidelines, Conventions, Idioms, and Patterns for Reusable .NET Libraries*, Addison-Wesley, published 2005.

[24] DeLine, Robert, *Code Talkers* in [75], 2010.

[25] Deursen, Steven van, and Mark Seemann, *Dependency Injection Principles, Practices, and Patterns*, Manning, 2019.

[26] Evans, Eric, *Domain-Driven Design: Tackling Complexity in the Heart of Software*, Addison-Wesley, published 2003.

[27] Feathers, Michael C., *Working Effectively with Legacy Code*, Prentice Hall, published 2004.

[28] Foote, Brian, and Joseph Yoder, *The Selfish Class* in [62], 1998.

[29] Forsgren, Nicole, Jez Humble, and Gen Kim, *Accelerate*, IT Revolution Press, 2018.

[30] Fowler, Martin, *CodeOwnership*, blog post at https://martinfowler.com/bliki/CodeOwnership.html, 2006.

[31] Fowler, Martin, *Eradicating Non-Determinism in Tests*, blog post at https://martinfowler.com/articles/nonDeterminism.html, 2011.

[32] Fowler, Martin, *Is High Quality Software Worth the Cost?*, blog post at https://martinfowler.com/articles/is-quality-worth-cost.html, 2019.

[33] Fowler, Martin, David Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, and Randy Stafford, *Patterns of Enterprise Application Architecture*, Addison-Wesley, 2003.

[34] Fowler, Martin, Kent Beck, John Brant, William Opdyke, and Don Roberts, *Refactoring: Improving the Design of Existing Code*, Addison-Wesley, 1999.

[35] Fowler, Martin, *StranglerFigApplication*, blog post at https://martinfowler.com/bliki/StranglerFigApplication.html, 2004.

[36] Freeman, Steve, and Nat Pryce, *Growing Object-Oriented Software, Guided by Tests*, Addison-Wesley, published 2009.

[37] Gabasova, Evelina, *Comparing F# and C# with dependency networks*, blog post at http://evelinag.com/blog/2014/06-09-comparing-dependency-networks, 2014.

[38] Gabriel, Richard P., *Patterns of Software. Tales from the Software Community*, Oxford University Press, 1996.

[39] Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, published 1994.

[40] Gawande, Atul, *The Checklist Manifesto: How to Get Things Right*, Metropolitan Books, 2009.

[41] Haack, Phil, *I Knew How To Validate An Email Address Until I Read The RFC*, blog post at https://haacked.com/archive/2007/08/21/i-knew-how-to-validate-an-email-address-until-i.aspx, 2007.

[42] Henney, Kevlin, tweet at https://twitter.com/KevlinHenney/status/3361631527, 2009.

[43] Herraiz, Israel, and Ahmed E. Hassan, *Beyond Lines of Code: Do We Need More Complexity Metrics?* in [75], 2010.

[44] Herzig, Kim Sebastian, and Andreas Zeller, *Mining Your Own Evidence* in [75], 2010.

[45] Hickey, Rich, *Simple Made Easy*, Strange Loop conference talk, 2011. A recording is available at https://www.infoq.com/presentations/Simple-Made-Easy.

[46] Hohpe, Gregor, and Bobby Woolf, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*, Addison-Wesley, published 2003.

[47] House, Cory, tweet at https://twitter.com/housecor/status/1115959687332159490, 2019.

[48] Howard, Michael, and David LeBlanc, *Writing Secure Code, Second Edition*, Microsoft Press, 2003.

[49] Humble, Jez, and David Farley, *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*, Addison-Wesley, published 2010.

[50] Hunt, Andy, and Dave Thomas, *The Pragmatic Programmer: From Journeyman to Master*, Addison-Wesley, 1999.

[51] Kahneman, Daniel, *Thinking, fast and slow*, Farrar, Straus and Giroux, 2011.

[52] Kay, Alan, and Andrew Binstock, *Interview with Alan Kay*, Dr. Dobb's, www.drdobbs.com/architecture-and-design/interview-with-alan-kay/240003442, July 10, 2012.

[53] Kerievsky, Joshua, *Refactoring to Patterns*, Addison-Wesley, published 2004.

[54] King, Alexis, *Parse, don't validate*, blog post at https://lexi-lambda.github.io/blog/2019/11/05/parse-don-t-validate, 2019.

[55] Kleppmann, Martin, *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*, O'Reilly, 2017.

[56] Lanza, Michele, and Radu Marinescu, *Object-Oriented Metrics in Practice: Using Software Metrics to Characterize, Evaluate, and Improve the Design of Object-Oriented Systems*, Springer, 2006.

[57] Levitt, Steven D., and Stephen J. Dubner, *Freakonomics—A Rogue Economist Explores The Hidden Side Of Everything*, William Morrow & Company, Revised and Expanded Edition 2006.

[58] Levitt, Steven D., and Stephen J. Dubner, *SuperFreakonomics: Global Cooling, Patriotic Prostitutes And Why Suicide Bombers Should Buy Life Insurance*, William Morrow & Company, 2009.

[59] Lippert, Eric, *Which is faster?*, blog post at https://ericlippert.com/2012/12/17/performance-rant, 2012.

[60] Martin, Robert C., and Micah Martin, *Agile Principles, Patterns, and Practices in C#*, Prentice Hall, published 2006.

[61] Martin, Robert C., *Clean Code: A Handbook of Agile Software Craftsmanship*, Prentice Hall, 2009.

[62] Martin, Robert C., Dirk Riehle, and Frank Buschmann (editors), *Pattern Languages of Program Design 3*, Addison-Wesley, 1998.

[63] Martin, Robert C., *The Sensitivity Problem*, blog post at http://butunclebob.com/ArticleS.UncleBob.TheSensitivityProblem, 2005?

[64] Martin, Robert C., *The Transformation Priority Premise*, blog post at https://blog.cleancoder.com/uncle-bob/2013/05/27/TheTransformationPriorityPremise.html, 2013.

[65] McConnell, Steve, *Code Complete, Second Edition*, Microsoft Press, 2004.

[66] Meszaros, Gerard, *xUnit Test Patterns: Refactoring Test Code*, Addison-Wesley, 2007.

[67] Meyer, Bertrand, *Object-oriented Software Construction*, Prentice Hall, 1988.

[68] Milewski, Bartosz, *Category Theory for Programmers*, originally a series of blog posts at https://bartoszmilewski.com/2014/10/28/category-theory-for-programmers-the-preface, 2014–2017. Also available as a print book, Blurb, 2019.

[69] Minsky, Yaron, *Effective ML*, recording of a lecture given at Harvard. The recording itself is available on YouTube at https://youtu.be/-J8YyfrSwTk, but you may instead prefer Yaron Minsky's web page that includes a bit of context: https://blog.janestreet.com/effective-ml-video, 2010.

[70] Neward, Ted, *The Vietnam of Computer Science*, blog post at http://blogs.tedneward.com/post/the-vietnam-of-computer-science, 2006.

[71] Norman, Donald A., *The Design of Everyday Things. Revised and Expanded Edition*, MIT Press, 2013.

[72] North, Dan, *Patterns of Effective Delivery*, Roots opening keynote, 2011. A recording is available at https://vimeo.com/24681032.

[73] Nygard, Michael T., *Release It! Design and Deploy Production-Ready Software*, Pragmatic Bookshelf, 2007.

[74] Nygard, Michael T., *DevOps: Tempo, Maneuverability, and Initiative*, DevOps Enterprise Summit conference talk, 2016. A recording is available at https://youtu.be/0rRWvsb8JOo.

[75] Oram, Andy, and Greg Wilson (editors), *Making Software: What Really Works, and Why We Believe It*, O'Reilly, 2010.

[76] O'Toole, Garson, *The Future Has Arrived – It's Just Not Evenly Distributed Yet*, online article on https://quoteinvestigator.com/2012/01/24/future-has-arrived, 2012.

[77] Ottinger, Tim, *Code is a Liability*, 2007. This was originally a blog post, but the original domain has since lapsed and been taken over by another entity. The blog post is still available via the Internet Archive at http://web.archive.org/web/20070420113817/http://blog.objectmentor.com/articles/2007/04/16/code-is-a-liability.

[78] Ottinger, Tim, *What's this about Micro-commits?*, blog post at https://www.industriallogic.com/blog/whats-this-about-micro-commits, 2021.

[79] Peters, Tim, *The Zen of Python*, 1999. Originally a mailing list post, it's long been available at https://www.python.org/dev/peps/pep-0020.

[80] Pinker, Steven, *How the Mind Works*, The Folio Society, 2013. I'm referring to my Folio Society edition, which, according to the colophon, "follows the text of the 1998 Penguin edition, with minor emendations." It was "first published by W.W. Norton in 1997."

[81] Pope, Tim, *A Note About Git Commit Messages*, blog post at https://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html, 2008.

[82] Poppendieck, Mary, and Tom Poppendieck, *Implementing Lean Software Development: From Concept to Cash*, Addison-Wesley, published 2006.

[83] Preston-Werner, Tom, *Semantic Versioning*, specification at https://semver.org. The root of the web site shows the latest version. As I'm writing in October 2020, the latest version is Semantic Versioning 2.0.0, which was published in 2013.

[84] Pyhäjärvi, Maaret, *Five Years of Mob Testing, Hello to Ensemble Testing*, blog post at https://visible-quality.blogspot.com/2020/05/five-years-of-mob-testing-hello-to.html, 2020.

[85] Rainsberger, J.B., *Integration Tests Are a Scam*, Agile 2009 conference talk, 2009. A recording is available at https://www.infoq.com/presentations/integration-tests-scam.

[86] Rainsberger, J.B., tweet at https://twitter.com/jbrains/status/167297606698008576, 2012.

[87] Reeves, Jack, *What Is Software Design?*, C++ Journal, 1992. If, like me, you don't have a copy of the C++ Journal lying around, you can find the article online. https://www.developerdotstar.com/mag/articles/reeves_design.html seems to have been stable for years. Also available as an appendix in [60].

[88] Ries, Eric, *The Lean Startup: How Constant Innovation Creates Radically Successful Businesses*, Portfolio Penguin, 2011.

[89] Robinson, Ian, Jim Webber and Emil Eifrem, *Graph Databases: New Opportunities for Connected Data. Second Edition*, O'Reilly, 2015.

[90] Scott, James C., *Seeing Like a State: How Certain Schemes to Improve the Human Condition Have Failed*, Yale University Press, 1998.

[91] Seemann, Mark, *10 tips for better Pull Requests*, blog post at https://blog.ploeh.dk/2015/01/15/10-tips-for-better-pull-requests, 2015.

[92] Seemann, Mark, *A heuristic for formatting code according to the AAA pattern*, blog post at https://blog.ploeh.dk/2013/06/24/a-heuristic-for-formatting-code-according-to-the-aaa-pattern, 2013.

[93] Seemann, Mark, *A red-green-refactor checklist*, blog post at https://blog.ploeh.dk/2019/10/21/a-red-green-refactor-checklist, 2019.

[94] Seemann, Mark, *Church-encoded Maybe*, blog post at https://blog.ploeh.dk/2018/06/04/church-encoded-maybe, 2018.

[95] Seemann, Mark, *CQS versus server generated IDs*, blog post at https://blog.ploeh.dk/2014/08/11/cqs-versus-server-generated-ids, 2014.

[96] Seemann, Mark, *Conway's Law: latency versus throughput*, blog post at https://blog.ploeh.dk/2020/03/16/conways-law-latency-versus-throughput, 2020.

[97] Seemann, Mark, *Curb code rot with thresholds*, blog post at https://blog.ploeh.dk/2020/04/13/curb-code-rot-with-thresholds, 2020.

[98] Seemann, Mark, *Devil's advocate*, blog post at https://blog.ploeh.dk/2019/10/07/devils-advocate, 2019.

[99] Seemann, Mark, *Feedback mechanisms and tradeoffs*, blog post at https://blog.ploeh.dk/2011/04/29/Feedbackmechanismsandtradeoffs, 2011.

[100] Seemann, Mark, *From interaction-based to state-based testing*, blog post at https://blog.ploeh.dk/2019/02/18/from-interaction-based-to-state-based-testing, 2019.

[101] Seemann, Mark, *Fortunately, I don't squash my commits*, blog post at https://blog.ploeh.dk/2020/10/05/fortunately-i-dont-squash-my-commits, 2020.

[102] Seemann, Mark, *Functional architecture is Ports and Adapters*, blog post at https://blog.ploeh.dk/2016/03/18/functional-architecture-is-ports-and-adapters, 2016.

[103] Seemann, Mark, *Repeatable execution*, blog post at https://blog.ploeh.dk/2020/03/23/repeatable-execution, 2020.

[104] Seemann, Mark, *Structural equality for better tests*, blog post at https://blog.ploeh.dk/2021/05/03/structural-equality-for-better-tests, 2021.

[105] Seemann, Mark, *Tautological assertion*, blog post at https://blog.ploeh.dk/2019/10/14/tautological-assertion, 2019.

[106] Seemann, Mark, *Towards better abstractions*, blog post at https://blog.ploeh.dk/2010/12/03/Towardsbetterabstractions, 2010.

[107] Seemann, Mark, *Visitor as a sum type*, blog post at https://blog.ploeh.dk/2018/06/25/visitor-as-a-sum-type, 2018.

[108] Seemann, Mark, *When properties are easier than examples*, blog post at https://blog.ploeh.dk/2021/02/15/when-properties-are-easier-than-examples, 2021.

[109] Shaw, Julia, *The Memory Illusion: Remembering, Forgetting, and the Science of False Memory*, Random House, 2017 (paperback edition; original published in 2016).

[110] Thomas, Neil, and Gail Murphy, *How Effective Is Modularization?* in [75], 2010.

[111] Tornhill, Adam, *Your Code as a Crime Scene: Use Forensic Techniques to Arrest Defects, Bottlenecks, and Bad Design in Your Programs*, Pragmatic Bookshelf, 2015.

[112] Tornhill, Adam, *Software Design X-Rays: Fix Technical Debt with Behavioral Code Analysis*, Pragmatic Bookshelf, 2018.

[113] Troy, Chelsea, *Reviewing Pull Requests*, blog post at https://chelseatroy.com/2019/12/18/reviewing-pull-requests, 2019.

[114] Webber, Jim, Savas Parastatidis, and Ian Robinson, *REST in Practice: Hypermedia and Systems Architecture*, O'Reilly, 2010.

[115] Weinberg, Gerald M., *The psychology of computer programming. Silver anniversary edition*, Dorset House Publishing, 1998.

[116] Williams, Laurie, *Pair Programming* in [75], 2010.

[117] Wlaschin, Scott, *Cycles and modularity in the wild*, blog post at https://fsharpforfunandprofit.com/posts/cycles-and-modularity-in-the-wild, 2013.

[118] Woolf, Bobby, *Null Object* in [62], 1997.