

## Список литературы

[Ammann2016] Paul Ammann and Jeff Offutt. *Introduction to Software Testing*. Cambridge University Press, Cambridge, United Kingdom, 2 edition, 2016.

**Abstract:** Extensively class-tested, this textbook takes an innovative approach to software testing: it defines testing as the process of applying a few well-defined, general-purpose test criteria to a structure or model of the software. It incorporates the latest innovations in testing, including techniques to test modern types of software such as OO, web applications, and embedded software. The book contains numerous examples throughout. An instructor's solution manual, PowerPoint slides, sample syllabi, additional examples and updates, testing tools for students, and example software programs in Java are available on an extensive website.

[Beck2002] Kent Beck. Preface. In *Test Driven Development By Example*, page xi. Addison-Wesley, Boston, MA, 1 edition, 2002.

[Briggs2021] Luke Briggs. Pepys, 2021.

**URL** <https://lukebriggs.dev/pepys>

[Cohn2005] Mike Cohn. The Purpose of Planning. In *Agile Estimating and Planning*, chapter 1, pages 3–9. Prentice Hall, Hoboken, New Jersey, 1 edition, 2005.

[Cunningham2008] Ward Cunningham. Ward Cunningham, inventor of Wiki, inventor of Fit, coinventor of eXtreme Programming. In Robert C. Martin, editor, *Clean Code: A Handbook of Agile Software Craftsmanship*, page 11. Pearson Education, Inc, Boston, MA, 2008.

[Hirsch1995] Walter L. Hirsch and Cristina Videira Lopes. Separation of Concerns. Technical report, College of Computer Science, Northeastern University, Boston, MA, 1995.

[Letouzey2016] Jean Louis Letouzey and Declan Whelan. Introduction to the Technical Debt Concept, 2016.

**URL** <https://www.agilealliance.org/wp-content/uploads/2016/05/IntroductiontotheTechnicalDebt.pdf>

**Abstract:** What is Technical Debt? Where does it comes from?

[Martin2008] Robert C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*. Pearson Education, Inc, Boston, MA, 2008.

[Peters2004] Tim Peters. PEP 20 – The Zen of Python, 2004.

**URL** <https://www.python.org/dev/peps/pep-0020/>

[Stevens1999] W. P. Stevens, G. J. Myers, and L. L. Constantine. Structured design. *IBM Systems Journal*, 38(2):231–256, 1999.

**Abstract:** Considerations and techniques are proposed that reduce the complexity of programs by dividing them into functional modules. This can make it possible to create complex systems from simple, independent, reusable modules. Debugging and modifying programs, reconfiguring I/O devices, and managing large programming projects can all be greatly simplified. And, as the module library grows, increasingly sophisticated programs can be implemented using less and less new code.

[Stroustrup2014] Bjarne Stroustrup. Make Simple Tasks Simple! In *CppCon 2014*, 2014.

[Thomas1999] David Thomas and Andrew Hunt. The Evils of Duplication. In *The Pragmatic Programmer*. Addison-Wesley, Reading, MA, 1999.

[UKGovernment2015] UK Government. BULK DATA TRANSFER:ADDITIONAL VALIDATION FOR CAS UPLOAD, 2015.

**URL** <https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachmen>

[bnf2020] *British National Formulary*. Pharmaceutical Press, London, 79 edition, 2020.

[haurant2004bbh] S. Haurant. Britain's borrowing hits £1 trillion. *The Guardian*, 29 July:16c, 2004.

[rothman.etal2008me] Kenneth J. Rothman, Sander Greenland, and Timothy L. Lash, editors. *Modern Epidemiology*. Lippincott Williams & Wilkins, Philadelphia, Pa., 3 edition, 2008.