

Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. Allen Downey, in his book *How To Think Like A Computer Scientist*, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages. Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL). Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly. There exist a lot of different approaches for each of those tasks. One approach popular for requirements analysis is Use Case analysis. Different programming languages support different styles of programming (called programming paradigms). It affects the aspects of quality above, including portability, usability and most importantly maintainability. Use of a static code analysis tool can help detect some possible problems. There are many approaches to the Software development process. The following properties are among the most important: In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. There exist a lot of different approaches for each of those tasks. It affects the aspects of quality above, including portability, usability and most importantly maintainability. It is usually easier to code in "high-level" languages than in "low-level" ones. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Scripting and breakpointing is also part of this process. Computer programmers are those who write computer software. Programmable devices have existed for centuries. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Use of a static code analysis tool can help detect some possible problems. It affects the aspects of quality above, including portability, usability and most importantly maintainability. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software.