Following a consistent programming style often helps readability. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Normally the first step in debugging is to attempt to reproduce the problem. Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Whatever the approach to development may be, the final program must satisfy some fundamental properties. 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 text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. It affects the aspects of quality above, including portability, usability and most importantly maintainability. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. 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). The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). However, readability is more than just programming style. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Code-breaking algorithms have also existed for centuries. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. 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. 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. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.