Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. 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. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. 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. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. 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. 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. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems. 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). One approach popular for requirements analysis is Use Case analysis. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. There are many approaches to the Software development process. It affects the aspects of quality above, including portability, usability and most importantly maintainability.