There are many approaches to the Software development process. 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. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Many applications use a mix of several languages in their construction and use. One approach popular for requirements analysis is Use Case analysis. Unreadable code often leads to bugs, inefficiencies, and duplicated code. However, readability is more than just programming style. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. 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. Also, specific user environment and usage history can make it difficult to reproduce the problem. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Programmable devices have existed for centuries. Ideally, the programming language best suited for the task at hand will be selected. 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. Ideally, the programming language best suited for the task at hand will be selected. 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. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.