After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. 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. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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. 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. Integrated development environments (IDEs) aim to integrate all such help. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Scripting and breakpointing is also part of this process. Unreadable code often leads to bugs, inefficiencies, and duplicated code. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Many applications use a mix of several languages in their construction and use. 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). Use of a static code analysis tool can help detect some possible problems. Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language. 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Normally the first step in debugging is to attempt to reproduce the problem. There are many approaches to the Software development process.