## Software Engineering Layers

Software engineering can be viewed as a layered technology. Various layers are listed below.

The **process layer**allows the development of software on time. It defines an outline for a set of key process areas that must be acclaimed for effective delivery of software engineering technology.

The **method layer**provides technical knowledge for developing software. This layer covers a broad array of tasks that include requirements analysis, design, coding, testing, and maintenance phase of the software development.

The **tools layer**provides computerized or semi-computerized support for the process and the method layer. Sometimes tools are integrated in such a way that other tools can use information created by one tool. This multi-usage is commonly referred to as **Computer-Aided Software Engineering (CASE).**CASE combines software, hardware, and software engineering [database](http://ecomputernotes.com/fundamental/what-is-a-database/advantages-and-disadvantages-of-dbms) to create software engineering analogous to **Computer-Aided Design (CAD) for**hardware. CASE helps in application development including analysis, design, code generation, and debugging and testing. This is possible by using CASE tools, which provide automated methods for designing and documenting traditional-structure programming techniques. For example, two prominent technologies using CASE tools are PC-based workstations and application generators that provide graphics-based interfaces to automate the development process.

## ****Explain Software Myths in Software Engineering****

## Management myths: The managers are often grasps at a belief in a software myth, same as a drowning person who grasps at a straw.

* **Members acquires all the information:** Generally, there is a myth that the members of the organization acquire all the information containing procedures, principles and standards. In reality, the developers don’t have information about all the established standards because they are often outdated, incomplete and unadaptable. Plus, there is a rare chance that the developer will follow all the standards.
* **Adding more people can reduce the time gap:** Another myth in the people is that more the number of programmers, lesser will be the time gap. If a project is behind the schedule, adding more manpower will further delay it because new workers will take more time to learn about the ongoing project.
* **The management can relax themselves by outsourcing its project**: If an organization outsource its software to a third party, it does not relieve the management of its duties. When the organization outsources the software project, they suffer invariably.

1. **Customers Myths:**The customers are encouraged by some marketing people in underestimating the difficulty of developing software.

* **Software is malleable as a result of which changes are easy to accommodate:**There is an enormous amount of labor required to have a change in the software after the release. It is not so easy to accommodate these changes.
* **To start coding, a general statement of need is enough:**The developers can’t read the customer’s mind and requires detailed descriptions of the requirements, in order to start coding.

1. **Developer Myths:**The software development art is becoming an engineering discipline, but there are lots of myths.

* **Once the code is delivered, the software can be called complete:**In reality, more than 60% of the efforts are expended after the delivery of the software to the user.
* **The software’s success depends on the product’s produced quality:**The project does not become successful on the quality of the programs because both the software configuration and documentation also play an important role in its success.
* **The unnecessary documentation is required in software engineering, which further slows down the growth rate of a project:**This myth is a no brainer because in reality, the proper documentation enhances the project’s quality and results in reduction of the rework. Also, this field is just about creating quality at all the level of the project.
* **The assessment of the software quality can be addressed after the execution of the program:**During any phase of the development process, the software’s quality can be measured just by applying the mechanism of quality assurance.
* **The product, which is delivered after the project’s completion can be called working program:** The deliverables of a successful project don’t only consist working program, but also the documentation which can guide the users about how to use the software.

So it was all about **Software Myths in Software Engineering.**If you have any doubt then please comment below.