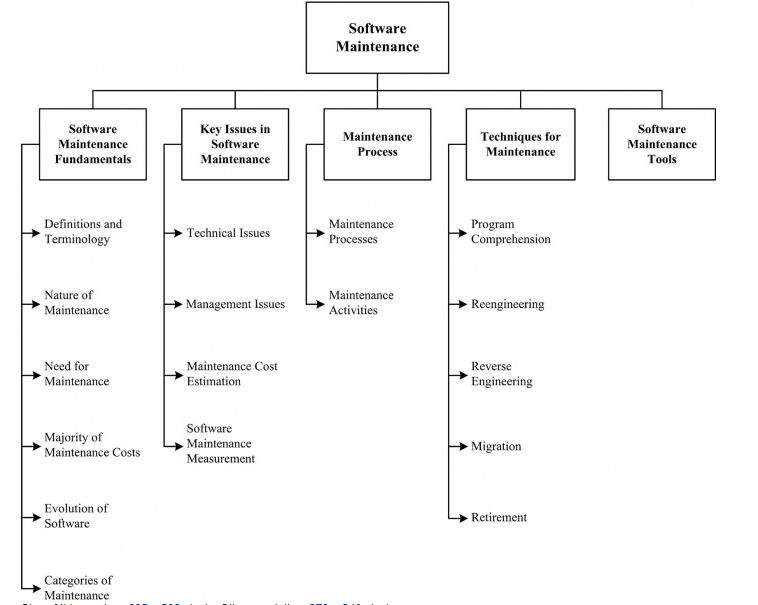
**SDLC Maintenance Phase:**



In the software life cycle, the maintenance phase is the last stage of the cycle. After software passes the design stage and is implemented, the maintenance phase of the software life cycle begins. Understanding the characteristics of the maintenance phase of the software life cycle allows individuals tasked with analyzing the performance of the software after deployment to correctly resolve issues that arise

The maintenance phase involves making changes to hardware, software, and documentation to support its operational effectiveness. It includes making changes to improve a system's performance, correct problems, enhance security, or address user requirements. To ensure modifications do not disrupt operations or degrade a system's performance or security, organizations should establish appropriate change management standards and procedures.

IEEE 14764 identifies the primary activities of software maintenance as process implementation, problem and modification analysis, modification implementation, maintenance review/acceptance, migration, and retirement.

**Why maintenance is important?**

After software is up and running, it often requires continuous maintenance. In general, software remains operational for an extended period of time after initial implementation and requires regular maintenance to ensure that the software operates continually at peak performance levels. During the maintenance phase of the software life cycle, software programmers regularly issue software patches to address changes in the needs of an organization, to correct issues relating to bugs in the software or to resolve potential security issues. Throughout the maintenance phase, designers address issues that are discovered to prevent any hindrance to the expected performance of the software or to add increased functionality to the software.

Maintenance must be performed in order to

* correct faults;
* improve the design;
* implement enhancements;
* interface with other software;
* adapt programs so that different hardware, software, system features, and telecommunications facilities can be used;
* migrate legacy software; and
* retire software.

Five key characteristics comprise the maintainer’s activities:

* maintaining control over the software’s day-to-day functions;
* maintaining control over software modification;
* perfecting existing functions;
* identifying security threats and fixing security vulnerabilities; and
* preventing software performance from degrading to unacceptable levels.

**Who performs maintenance?**

Software maintenance tasks are typically performed by the designers of the software to address issues discovered after deployment of the software program. As software developers address the issues found in software programs, it falls to system operators to install the released patches. Most commercially available software programs are regularly updated using downloads available through the developer's online support site. System operators download updates as they are released and install these to ensure continued performance of the software in accordance with the original design parameters.

**Length of maintenance phase:**

The maintenance phase lasts the longest of all phases of the software life cycle. The primary reason that the maintenance phase last longest is the life of the software. Most software programs remain in operation until they become obsolete due to changes in a user's needs or until the cost of maintaining the software becomes prohibitive to future use of the software.

Software maintenance sustains the software product throughout its life cycle (from development to operations). Modification requests are logged and tracked, the impact of proposed changes is determined, code and other software artifacts are modified, testing is conducted, and a new version of the software product is released. Also, training and daily support are provided to users. The term maintainer is defined as an organization that performs maintenance activities. In this KA, the term will sometimes refer to individuals who perform those activities, contrasting them with the developers.

*Categories of Maintenance*

[[1](http://swebokwiki.org/Chapter_5:_Software_Maintenance#References), c3, c6s2] [[2](http://swebokwiki.org/Chapter_5:_Software_Maintenance#References), c3s3.1]

Three categories (types) of maintenance have been defined: corrective, adaptive, and perfective. IEEE 14764 includes a fourth category–preventative.

* Corrective maintenance: reactive modification (or repairs) of a software product performed after delivery to correct discovered problems. Included in this category is emergency maintenance, which is an unscheduled modification performed to temporarily keep a software product operational pending corrective maintenance.
* Adaptive maintenance: modification of a software product performed after delivery to keep a software product usable in a changed or changing environment. For example, the operating system might be upgraded and some changes to the software may be necessary.
* Perfective maintenance: modification of a software product after delivery to provide enhancements for users, improvement of program documentation, and recoding to improve software performance, maintainability, or other software attributes.
* Preventive maintenance: modification of a software product after delivery to detect and correct latent faults in the software product before they become operational faults.

## **Cost of Maintenance**

Reports suggest that the cost of maintenance is high. A study on estimating software maintenance found that the cost of maintenance is as high as 67% of the cost of entire software process cycle.



On an average, the cost of software maintenance is more than 50% of all SDLC phases.

**References:**

<http://swebokwiki.org/Chapter_5:_Software_Maintenance>

<https://ithandbook.ffiec.gov/it-booklets/development-and-acquisition/development-procedures/systems-development-life-cycle/maintenance-phase.aspx>