THALES
Building a future we can all trust

The 4 Types of Software Maintenance

What is Software Maintenance?

Software maintenance is the process of changing, modifying, and updating software to keep up with customer needs. Software maintenance is done after the product has launched for several reasons including improving the software overall, correcting issues or bugs, to boost performance, and more.

Software maintenance is a natural part of SDLC (software development life cycle). Software developers don’t have the luxury of launching a product and letting it run, they constantly need to be on the lookout to both correct and improve their software to remain competitive and relevant.

Using the right software maintenance techniques and strategies is a critical part of keeping any software running for a long period of time and keeping customers and users happy.

Why is software maintenance important?

Creating a new piece of software and launching it into the world is an exciting step for any company. A lot goes into creating your software and its launch including the actual building and coding, licensing models, marketing, 'are must be able to adapt to the times.

erly. As technology is changing at the speed of light, software must ıds.

ıce?

The four types are:
Corrective Software Maintenance
Preventative Software Maintenance
Perfective Software Maintenance
Adaptive Software Maintenance

Corrective Software Maintenance

Corrective software maintenance is the typical, classic form of maintenance (for software and anything else for that matter). Corrective software maintenance is necessary when something goes wrong in a piece of software including faults and errors. These can have a widespread impact on the functionality of the software in general and therefore must be addressed as quickly as possible.

Many times, software vendors can address issues that require corrective maintenance due to bug reports that users send in. If a company can recognize and take care of faults before users discover them, this is an added advantage that will make your company seem more reputable and reliable (no one likes an error message after all).

Preventative Software Maintenance

Preventative software maintenance is looking into the future so that your software can keep working as desired for as long as possible.

This includes making necessary changes, upgrades, adaptations and more. Preventative software maintenance may address small issues which at the given time may lack significance but may turn into larger problems in the future. These are called latent faults which need to be detected and corrected to make sure that they won't turn into effective faults.

Perfective Software Maintenance

As with any product on the market, once the software is released to the public, new issues and ideas come to the surface. Users may see the need for new features or requirements that they would like to see in the software to make it the best tool available for their needs. This is when perfective software maintenance comes into play.

Perfective software maintenance aims to adjust software by adding new features as necessary and removing features that are irrelevant or not effective in the given software. This process keeps software relevant as the market, and user needs, change.

Adaptive Software Maintenance

h the changing technologies as well as policies and rules regarding n changes, cloud storage, hardware, etc. When these changes are r to properly meet new requirements and continue to run well.

cess

according to the type of maintenance and the software maintenance plan in place.

Most software maintenance process models include the following steps:

1. Identification & Tracing – The process of determining what part of the software needs to be modified (or maintained). This can be user-generated or identified by the software developer itself depending on the situation and specific fault.
2. Analysis – The process of analyzing the suggested modification including understanding the potential effects of such a change. This step typically includes cost analysis to understand if the change is financially worthwhile.
3. Design – Designing the new changes using requirement specifications
4. Implementation – The process of implementing the new modules by programmers.
5. System Testing – Before being launched, the software and system must be tested. This includes the module itself, the system and the module, and the whole system at once.
6. Acceptance Testing- Users test the modification for acceptance. This is an important step as users can identify ongoing issues and generate recommendations for more effective implementation and changes.
7. Delivery – Software updates or in some cases new installation of the software. This is when the changes arrive at the customers.

Software Maintenance Cost

The cost of software maintenance can be high. However, this doesn't negate the importance of software maintenance. In certain cases, software maintenance can cost up to two-thirds of the entire software process cycle or more than 50% of the SDLC processes.

The costs involved in software maintenance are due to multiple factors and vary depending on the specific situation. The older the software, the more maintenance will cost, as technologies (and coding languages) change over time. Revamping an old piece of software to meet today's technology can be an exceptionally expensive process in certain situations.

In addition, engineers may not always be able to target the exact issues when looking to upgrade or maintain a specific piece of software. This causes them to use a trial and error method, which can result in many hours of work.

software maintenance costs. These include optimizing the top of
ping, and functional programming.

on maintenance projects for older models, software companies
nsideration. Without maintenance, any software will be obsolete

All software companies should have a specific strategy in place to tackle software maintenance in an effective and complete manner.

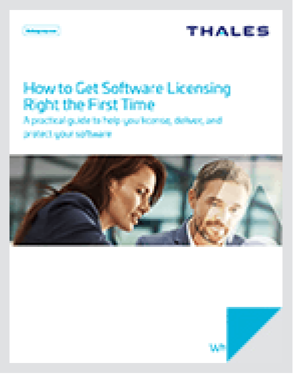
Documentation is one important strategy in software development. If software documentation isn't up to date, upgrading can be seemingly impossible. The documentation should include info about how the code works, solutions to potential problems, etc.

QA is also an important part of a software maintenance plan. While QA is important before an initial software launch, it can also be integrated much earlier in the process (as early as the planning stage) to make sure that the software is developed correctly and to give insight into making changes when necessary.

Using Thales to boost your software maintenance techniques today

Having a software maintenance plan in place is the first step to effective software maintenance. In order to do so, you need the right technology set up that works with your software.

Thales' software licensing and management platforms give you easy access to see what is happening with your software in real-time, on the ground and to keep in consistent contact with your customers.



THALES

How to Get Software Licensing Right the First Time

A practical guide to help you license, deliver, and protect your software

Navigate The Process of Licensing, Delivering, and Protecting Your Software

How to Get Software Licensing Right The First Time - White Paper Check out our practical guide to navigating the process of licensing, delivering, and protecting your software. Discover the importance of optimizing your …s with the…

Read More

Flexible Software Licensing helps Reduce Costs - Eocortex Case Study

How Eocortex Reduced Costs and Improved User Experience with Sentinel Even Security Surveillance Software Needs Flexible Licensing Security Eocortex's innovative and customizable security solutions include some of the most advanced features on the global video surveillance...

Read More

Maximize the value of your software

Contact a Sales Specialist

THALES
Building a future we can all trust

cpl.thalesgroup.com

Visit our parent site at
www.thalesgroup.com

Products
Data Protection
Access Management & Authentication
Software Monetization

Solutions
By Use Case
By Industry
By Compliance

Partners
Find a Partner
Become a Partner
Partner Portal Login

Resources
Resources Library
Blog
Podcasts

Support
Customer Support
Training Services

About
Contact Us
India E-Waste Program
Newsroom