ANALYSIS OF SOFTWARE MAINTENANCE PHASES

Shirinbaji Umudova*

*Azerbaijan National Academy of Sciences, Institute of Information Technology, Azerbaijan

ABSTRACT: In the software engineering software development life cycle is a sequence of steps to prepare and develop software. Software maintenance is essential in these stages. After the software is delivered, there are some changes to the software that are required for some reason. These changes may be related with software bug fixes, software upgrade, and so on. The main purpose of software maintenance is to eliminate these problems. The article deals with these issues. In this article software development life cycle models and phases have been investigated and its maintenance phase has been reviewed. Problems that arise during software maintenance have been identified.

Keywords: Software, Lifecycle, Software Maintenance, Models.

1. INTRODUCTION

The main purpose of the software lifecycle is to develop high quality, efficient software that responds user requirements. Software life cycle is divided into different phases in order to organize software quickly and efficiently. The number of these phases depends on what software is used for and purpose. Usually this number varies between five and seven [1].

There are different models of software life cycle, and the choice of the model depends on the project. Each software engineer should have enough knowledge in this area to select the appropriate software lifecycle based on the content and requirements of the project [2]. The latest and most important phase of software lifecycle in these models is software maintenance. The latest and most important phase of software lifecycle in these models is software maintenance.

The main goal in maintenance of software is to maintain work ability of software over time. Software maintenance by a wide range of activities incorporates the correctness of the errors that arise after delivering the software to the customer, increasing the working capacity, deleting and optimizing outdated functions. Software maintenance costs make up about 20-25% of the total cost of the software.

The main purpose of the article is to investigate the software maintenance process and its key issues, to provide information on life cycle models for software.

2. SOFTWARE DEVELOPMENT LIFE CYCLE MODELS AND PHASES

2.1. Software Development Life Cycle Models

Software development life cycle has different models. The most commonly used are [3]:

1. Waterfall Model - This model was first proposed by Herbert D. Benington in 1956 and is the most widely used model. The model consists of different phases. The current phase in the development process should only be completed after the previous phase has been completed [4].

2. V-shaped model - The V-shaped model is similar to the waterfall model. In this model, one phase is passed to the other after completion. The distinctive feature of this model is its constant testing [5].

3. Spiral Model - This model was introduced by American software engineer Barry W. Boehm. The phases of this model are accompanied by iterations. This model is suitable only for large projects [6].
4. Bing Bang Model - This model does not require any planning. The developer himself analyzes, codifies, and develops the project as he understands. There is no test group and no formal test is required. This can lead to the project's failure. This model is only used for small projects.

It is important to select a suitable model for the successful completion of the project. The appropriate model is selected mainly requirements, system complexity, project size, price, and so on. Each model has its own unique phases.

2.2. Software Development Life Cycle

Software development life cycle consists of the following phases [7]:

1. Requirement gathering – In this phase information about the software to be created, the aim that it will be used for is collected from the user. Requirements are divided into three groups: user requirements, system requirements and functional requirements. These requirements are shown in Figure 1.

![Figure 1. Software requirements](source: Author)

User Requirements: One of the most important and difficult phase of software development is to identify user requirements. The reason is that in some cases the inability of the user to fully co-ordinate his own needs and desires is incomplete, inaccurate and contradictory.

System requirements: System requirements are based on user requirements. Requirements must have features such as accuracy, availability, compliance, effectiveness, privacy, security, and quality.

Functional requirements: Functional requirements are functions designed to perform system components. Calculations, technical data, data manipulation and processing, and other special features refer to functional requirements.

2. Software design - This phase consists of software design in accordance with all requirements.
3. Coding - This phase is a programming phase. In this phase the selected programming language is used to write the program code and its efficient operation is ensured.
4. Testing - Testing phase is carried out by testing experts. This phase starts after the coding phase is completed. Quickly detecting errors and fixing them is one of the key conditions for developing secure software.
5. Deployment - In this phase the software is installed on a user's computer and tested in real-environment.
6. Maintenance - This phase covers the next phase after delivering software.

3. SOFTWARE MAINTENANCE

The final phase of the software lifecycle - the maintenance phase takes an important place. The main purpose of software maintenance is to solve the problems mentioned in Figure 2 [8].

![Figure 2. Software maintenance issues](source: Author)
**Bug fixing** - Software bug fixes are one of the key priorities for the program to work smoothly. At this time, errors in the code are searched, found and corrected [9]. Errors may be in technical support, operating system or any part of the software. Correcting these errors should be handled so that the existing software functions are not damaged.

**Capability enhancement** - This will improve functionality of programs to ensure compatibility with changing market environment.

**Removal of outdated functions** - Outdated functions are useless. These functions damage the effectiveness of the process by taking an important place in the solution process. When software maintenance, such codes are removed from the program and replaced by new advanced tools and technologies.

**Performance improvement** - To develop the system, programmers identify and solve problems through tests.

### 3.1. Software Maintenance Categories

Software maintenance is grouped into four categories [10]. These categories are shown in Figure 3.

**Figure 3.** Software maintenance categories

**Corrective** - correction of program errors. Software errors can be of different types. These are errors related to coding errors, design errors and requirements. Coding errors are corrected by programmers. The mistakes about the requirements are even more important. Because the software is built to meet certain requirements and the system needs to be re-created when these requirements are incorrect.

**Perfective** - add or modify functionality to the system. Changes are made to this category to make a more efficient and functional system in accordance with changing requirements.

**Preventive** - This category aims to increase the duralibility or reliability of software to prevent future problems.

**Adaptive** - Adapt software to different operating environments. It is one of the important factors that the software is adapting to different environments during its lifecycle. The main purpose of this category is to ensure that software is adapted to different environments.

During the software maintenance, the categories we mentioned above appear in the percentage shown in Figure 4.

**Figure 4.** Percentage of software maintenance categories

**Source:** Author
3.2. Software Maintenance Phases

The software maintenance process consists of seven different stages.

**Change management** - At this stage, the user determines the changes. Also, the category of change is determined.

**Analysis** - The scope of each approved change is determined and a plan is prepared to make changes to the software.

**Design** - Changing the system is actually planned at this stage. It contains information about the existing system, project documentation, database and existing software, analysis phase.

Its purpose is to compile changes all the categories.

**Implementation** - This stage includes coding, testing, custom code assimilation, integration, analysis and so on.

**Regression Testing** - The regression test is performed to make sure that there are no errors in the system and that there is no error in the software after this test is performed [11].

**Acceptance Testing** - The main purpose of this test is to check whether all features of the program are in line with the requirements for change.

**Delivery** - The software is delivered to the user after the acceptance test is completed. The system's final test is performed by the customer after the system is submitted. Steps to software maintenance are shown in Figure 5.

![Figure 5. Stages of the software maintenance](image)

Source: Author

Various models are used for solving problems in the software. These models offer different approaches and techniques to facilitate the maintenance process. The most commonly used models are [12]:

**Quick-Fix Model**: The main purpose of this model is to identify the problem and resolve it as soon as possible. The advantage of this model is carrying out the work with little expenses.

**Boehm Model**: The basis of this model is economic models and principles. The use of economic models helps to better understand the problem and increase productivity in the maintenance process.

**Iterative enhancement model**: This model looks at the program as a process of repetition of the changes throughout the life cycle. The model consists of three stages. First, the system should be analyzed. Subsequently the proposed changes are categorized. Finally, changes are made.

3.3. Problems with Software Maintenance

A number of problems arise during the software maintenance [13]. These problems are the size of the database, the duration of the system, the budget for system maintenance, the size of the system, the number of staff, and so on. The size of the database is measured by the number of data files and the number of characters in the database. Depending on the client's requirement, certain database changes are made. Making changes to the database that's large is not easy. The size of the system is determined by the number of software modules found in the program. Larger systems are more complicated and their maintenance is more important. After the software is delivered, changes are made to the system at the request of the customer, which is performed by the team for the development of the software. If the organization hand over these works to a new programmer or worker, additional training is required by the organization, which will result in additional time and expense.
4. CONCLUSION

Proper organization of the software accompanying process is one of the key factors for developing quality software. Here are the life cycle models of software and its phases have been analyzed. Software accompaniment and its phases have been studied, and problems emerging at that time have been considered. It has been determined that, with timely detection of errors and other problems, it is possible to obtain more reliable software.

REFERENCES


