SOFTWARE MAINTENANCE AND EVOLUTION

Assoc. Prof. Dr. Mehmet Akif Çifçi

What is Software Maintenance?

- It is unthinkable to think of a software system that will not need to change over time. Changes in the needs of users or customers need to be reflected in the system. In addition, the operating conditions of the system may change due to a new hardware or software infrastructure. Of course, unnoticed errors can be detected during testing and need to be fixed.

Why Software Maintenance?

- Software maintenance is a process that involves periodic, ongoing and sometimes continuous management of software to ensure that it continues to function properly and efficiently. It may be required for the following:
  - Error Correction: In maintenance management, bug fixing is a priority to keep the software running smoothly. It involves looking for and fixing bugs in the code. Problems can arise in hardware, in operating systems, and possibly in any part of the software. This means that available updates should be made without damaging the rest of the software's functionality.
  - Improving Design

Why Software Maintenance?

- Implementation of Improvement: This involves an improvement in features and functionality to align solutions with the changing market environment.
- Interfacing with other systems (e.g. databases).
- Hosting programs (for example, a database system) so that different hardware, software, system features and telecommunications facilities can be used.
- Removing obsolete features (e.g. uninstalling outdated software). Unwanted functions are useless; they also take up space in the solution.

Why Software Maintenance is Needed?

- Ensuring service continuity: Keeping the system operational and accessible,

- Providing mandatory update support: Meeting requests due to changes in the law and the platform on which the software runs,

Why Software Maintenance is Needed?

- Providing for changes in user requirements: Meeting users' requests for increased functionality and changes in requirements,

- The need for future maintenance: The need to restructure code or database and update documentation due to shortcuts in software development for commercial and financial reasons.

What are the Types of Care?

Maintenance is not only about fixing bugs but also includes various tasks that need to be done after the delivery of the product. According to ISO/IEC 14764-2006, 4 types of software maintenance are defined.

- Corrective maintenance
- Adaptive maintenance
- Remedial care
- Preventive care

Corrective Maintenance

The work to investigate the source of the detected errors and to eliminate the error is called corrective maintenance. To avoid too frequent corrective maintenance;

- Adopt a strong testing practice,
- Develop high quality code,
- Focus on the correct implementation of the design specification,
- Develop your ability to anticipate problems.

Adaptive Maintenance

- It is the adaptation of software to a new working environment.
- The platform on which the software runs may be changing due to technology, laws, policies, rules, operating system, etc. Keep in mind that if you are changing the environment your software lives in, it will trigger changes in other parts of your software.
- You need to carry out adaptive maintenance quickly because delay can result in not being able to carry out proper maintenance later on, which is an expensive approach.

Remedial Care

- Maintenance works such as adding new functions and features to the system and increasing performance. For example;
  - Speed optimization,
  - Improvement in user interfaces,
  - Improvements in software availability,
  - Improving software functionality,
  - Improved software performance.
- It is often much more costly to add new functionality to a system after it is up and running than it is to add the same functionality while it is still in development.

Preventive Maintenance

- Preventive maintenance is the precautionary actions that can be taken to increase maintainability and reliability in order to better adapt the software to changes that may be applied in the future. For example;
  - Updating documents: Updating the document according to the current state of the system.
  - Optimizing code: Modifying code to make programs execute faster or to use storage space efficiently.
  - Code restructuring: Transforming the structure of the program by reducing the source code, making it easier to understand.

Maintenance Stages

When a maintenance task arises for software in the maintenance phase, a standard process should be followed by the software developer. We can summarize the maintenance phase as follows.

- Problem Definition Phase
- Problem Analysis Phase
- Design Phase
- Implementation Phase
- System Test Phase
- Acceptance Test Phase
- Delivery Phase

Maintenance Activities

It is a field of task- and process-based relationships and roles that utilize knowledge from management, communication, software and computer science. Maintenance activities;

- Related to managing maintenance and software configuration; administration
- For testing changes to its software; audit
- Related to making sure that changes do not damage the whole product; quality
- Research to identify and analyze the impacts that require change

Care Request

In order to create a qualified maintenance phases in a software project, the following

situations need to be taken into consideration.

- Maintenance requests are always handled as part of a formal change control process

must be taken,

- Maintenance must be carried out quickly,
- Accurate needs analysis should be made for maintenance requests (labor, cost, duration, etc.),
- The status of care requests should be systematically reviewed,
- Tools should be used to increase productivity and facilitate maintenance.

Maintenance Issues

Maintenance problems can be expressed as follows;

- Maintenance becomes more difficult the higher the version of the software.
- Following the exact process by which software is developed is time and labor intensive.
- It often takes time to understand code written by someone else
- Documentation may be inadequate and incomplete
- Ability to modify most of the software

Change Engineering

Since the aim of Change Engineering is to have a positive impact on performance, it is an important concept for the human resources department in an organization. Change engineering, as the name suggests, is a concept about change. It is a system established to produce a holistic solution to the existing change demands in the organization. The highest efficiency can be obtained when this system proceeds in a planned and organized manner.

Change Engineering Features

- Constantly questioning
- Do not act on assumptions
- No fixed values
- It does not waste time with the existing
- Subjects the existing to real and definite changes
- Creates new methods
- Aim for effective transformations in performance
- Relates to the whole process

What is Reorganization?

- Refactoring software is the process of restructuring or improving existing code while preserving its external behavior. In other words, it involves optimizing the internal structure of your codebase without affecting its functioning. Refactoring is an important aspect of software development that helps improve the readability, maintainability and performance of code by making it cleaner, modular and more efficient.

Code Reorganization

- Improves code quality
- Reduces technical debt
- Improves maintainability and scalability
- Makes debugging and testing easier
- Increases developer productivity
- Allows new team members to be brought on board

Software Refactoring Techniques

- Rename Method: Rename methods and variables to make their purpose clearer. Meaningful names make the code easier to understand and maintain.

- Subtraction Method: Reorganize long or complex methods by breaking them down into smaller, more manageable functions that perform specific tasks. This improves code readability and maintainability.

Software Refactoring Techniques

- Replace Magic Numbers with Constants: Replace hard-coded values known as "magic numbers" with meaningful constant names to improve code readability and reduce the chance of errors.
- Extract Common Code: Identify common patterns or code sections that are repeated and extracted into separate reusable functions to reduce redundancy and increase sustainability.
- Move Method: Refactor methods defined in the wrong class or module by moving them to the appropriate location, improving code structure and maintainability.

System Management

- System administration encompasses the tasks and responsibilities necessary to ensure the efficient and secure operation of a computer system or network. These tasks include installation and management of hardware and software, performance monitoring and troubleshooting, security and backup.

System Management Concepts

- Hardware: The physical components that make up a computer system.
- Software Programs that allow the computer system to function.
- Operating system: Software that enables communication between hardware and software in a computer system.
- Network: A system that allows computers and other devices to communicate with each other.
- Data: Information stored in a computer system.
- Security: The protection of a system, network or data environment from unauthorized access, use, disclosure, modification or destruction.

Doc. Dr. Mehmet Akif Cifci

- Technical University of Vienna (Austria)
- University of Klaipeda (Lithuania)
- Bandirma Seventeen Eylul University

To Follow and Connect

https://github.com/themanoftalent
https://www.linkedin.com/in/themanoftalent/
https://www.researchgate.net/profile/Mehmet-Akif-Cifci