

SWE-417 SOFTWARE REENGINEERING

**Software Engineering Department
Sir Syed University of Engineering &
Technology**

Week No. 5

CONTENTS

- Taxonomy of Software Maintenance And Evolution
 - Intention-based Classification of S/W Maintenance
 - Activity-based Classification of S/W Maintenance
 - Evidence-based Classification of S/W Maintenance
 - Categories of Maintenance Concepts

GENERAL IDEA

- In circa 1972, R. G. Canning in his landmark article: “The Maintenance ‘Iceberg’,” discussed the problems of software maintenance.
- Practitioners took a narrow view of maintenance as :
 - correcting errors, and
 - enhancing the functionalities of the software.

GENERAL IDEA

- The ISO/IEC 14764 standard defines software maintenance as:
 - “... the totality of activities required to provide cost-effective support to a software system. Activities are performed during the pre-delivery stage as well as the post-delivery stage.”
 - Post-delivery activities includes changing software, providing training, and operating a help desk.
 - Pre-delivery activities include planning for post-delivery operations.

INTENTION-BASED CLASSIFICATION OF S/ W MAINTENANCE

- Maintenance activities are divided into four groups:
 - **Corrective maintenance**
 - **Adaptive maintenance**
 - **Perfective maintenance**
 - **Preventive maintenance**

INTENTION-BASED CLASSIFICATION OF S/ W MAINTENANCE

- **Corrective maintenance:** The purpose of corrective maintenance is to correct failures, processing failures and performance failures.
 - Examples of corrective maintenance:
 - A program producing a wrong output is an example of processing failure.
 - Similarly, a program not being able to meet real-time requirements is an example of performance failure.
 - Corrective maintenance is a reactive process, which means that corrective maintenance is performed after detecting defects with the system.

INTENTION-BASED CLASSIFICATION OF S/ W MAINTENANCE

- **Adaptive maintenance:** The purpose of adaptive maintenance is to enable the system to adapt to changes in its data environment or processing environment.
 - This process modifies the software to properly interface with a changing or changed environment.
 - Adaptive maintenance includes system changes, additions, deletions, modifications, extensions, and enhancements to meet the evolving needs of the environment in which the system must operate.
 - Examples of Adaptive maintenance are:
 - Changing the system to support new hardware configuration;
 - Converting the system from batch to on-line operation; and
 - Changing the system to be compatible with other applications.

INTENTION-BASED CLASSIFICATION OF S/ W MAINTENANCE

- **Perfective maintenance:** The purpose of perfective maintenance is to make a variety of improvements, namely, user experience, processing efficiency, and maintainability.
 - Examples of perfective maintenance are:
 - The program outputs can be made more readable for better user experience;
 - The program can be modified to make it faster, thereby increasing the processing efficiency; and
 - The program can be restructured to improve its readability, thereby increasing its maintainability.
 - Activities for perfective maintenance include restructuring of the code, creating and updating documentations, and tuning the system to improve performance.
 - It is also called “reengineering”.

INTENTION-BASED CLASSIFICATION OF S/ W MAINTENANCE

- **Preventive maintenance:** The purpose of preventive maintenance is to prevent problems from occurring by modifying software products. Basically, one should look ahead, identify future risks and unknown problems, and take actions so that those problems do not occur.
 - Preventive maintenance is very often performed on safety critical and high available software systems.
 - The concept of “**software rejuvenation**” is a preventive maintenance measure to prevent, or at least postpone, the occurrences of failures (crash) due to continuously running the software system.
 - It involves occasionally terminating an application or a system, cleaning its internal state, and restarting it.
 - Rejuvenation may increase the downtime of the application; however, it prevents the occurrence of more severe failures.

ACTIVITY-BASED CLASSIFICATION OF SOFTWARE MAINTENANCE

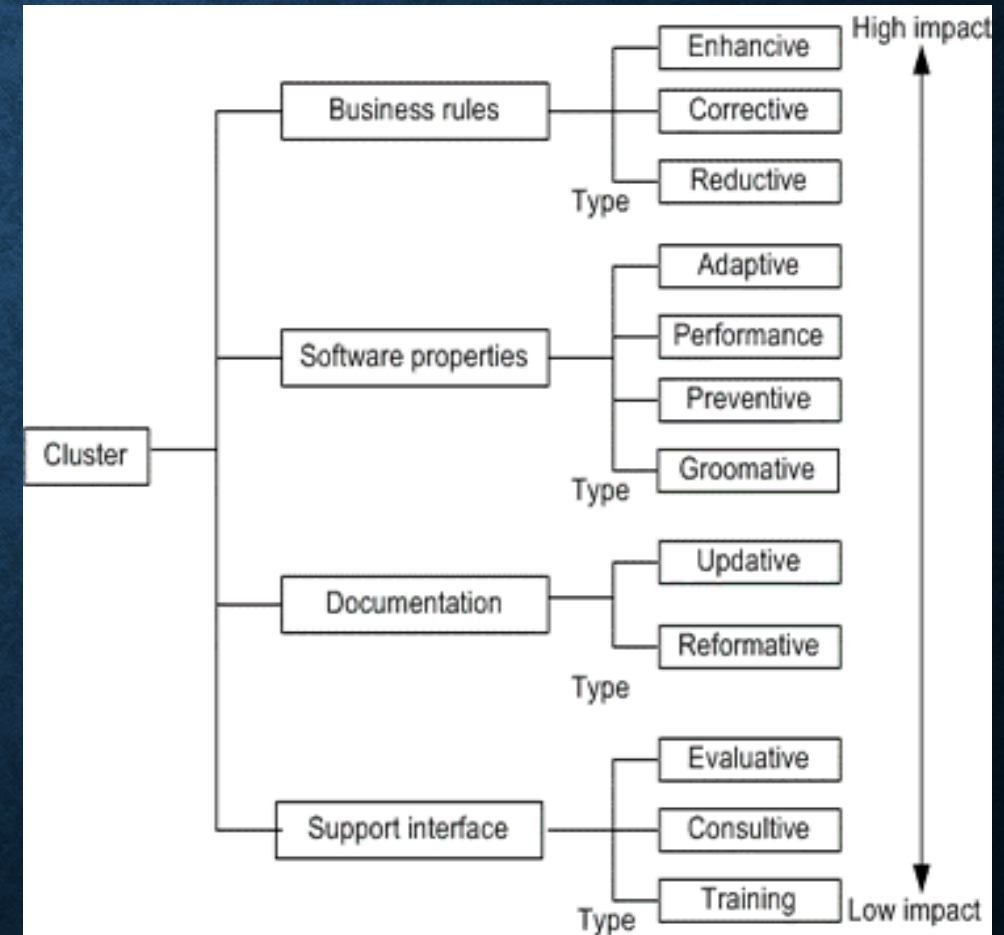
- In the intention-based classification of maintenance activities, the intention of an activity depends upon the reason for the change. The authors classify the maintenance modification activities into two categories: corrections and enhancements.
- **Corrections:** Activities in this category are designed to fix defects in the system, where a defect is a discrepancy between the expected behavior and the actual behavior of the system.

ACTIVITY-BASED CLASSIFICATION OF SOFTWARE MAINTENANCE

- **Enhancements:** Activities in this category are designed to effect changes to the system. This category of activities is further divided into three subcategories as follows:
 - Enhancement activities that modify some of the existing requirements implemented by the system;
 - Enhancement activities that add new system requirements;
 - Enhancement activities that modify the implementation without changing the requirements implemented by the system.

EVIDENCE-BASED CLASSIFICATION OF SOFTWARE MAINTENANCE

- Twelve types of maintenance activities were grouped into four clusters.
 - Business Rules
 - Software Properties
 - Documentation
 - Support Interface



Groups or clusters and their types

EVIDENCE-BASED CLASSIFICATION OF SOFTWARE MAINTENANCE

- Business Rules
 - **Enhancive:** Ordinary activities in this type are adding and modifying business rules to enhance the system's functionality available to the customer, and adding new data flows into or out of the software.
 - **Corrective:** Ordinary activities in this type are correcting identified bugs, adding defensive programming strategies, and modifying the ways exceptions are handled.
 - **Reductive:** Ordinary activities in this type drop some data generated for the customer, decreasing the amount of data input to the system, and decreasing the amount of data produced by the system.

EVIDENCE-BASED CLASSIFICATION OF SOFTWARE MAINTENANCE

- Software Properties
 - **Adaptive:** Ordinary activities in this type port the software to a different execution platform, and increase the utilization of Commercial off-the shelf (COTS) software components.
 - **Performance:** Activities in performance type produce results that impact the user. Those activities improve system up time and replace components and algorithms with faster ones.
 - **Preventive:** Ordinary activities in this type perform changes to enhance maintainability, and establish a base for making a future transition to an emerging technology.
 - **Groomative:** Ordinary activities in this type are substituting components and algorithms with more efficient and simpler ones, modifying the conventions for naming data, changing access authorizations, compiling source code, and doing backups.

EVIDENCE-BASED CLASSIFICATION OF SOFTWARE MAINTENANCE

- Documentation
 - **Update:** Ordinary activities in this type are substituting out-of-date documentation with up-to-date documentation, making semi-formal, say, in UML to document current program code, and updating the documentation with test plans.
 - **Reformative:** Ordinary activities in this type to improve the readability of the documentation, make the documentation consistent with other changes in the system, prepare training materials, and add entries to a data dictionary.

EVIDENCE-BASED CLASSIFICATION OF SOFTWARE MAINTENANCE

- Support Interface
 - **Evaluative:** In this type, common activities include reviewing the program code and documentations, examining the ripple effect of a proposed change, designing and executing tests, examining the programming support provided by the operating system, and finding the required data and debugging.
 - **Consultive:** In this type, cost and length of time are estimated for maintenance work, personnel run a help desk, customers are assisted to prepare maintenance work requests, and personnel make expert knowledge about the available resources and the system to others in the organization to improve efficiency.
 - **Training:** This means training the stakeholders about the implementation of the system.

EVIDENCE-BASED CLASSIFICATION OF SOFTWARE MAINTENANCE

- Modifications performed, detected, or observed on four aspects of the system being maintained are used as the criteria to cluster the types of maintenance activities:
 - The whole software;
 - The external documentation;
 - The properties of the program code; and
 - The system functionality experienced by the customer.
- Evidence of changes to those entities is gathered by comparing the appropriate portions of the software before the activity with the appropriate parts after the execution of the activity.

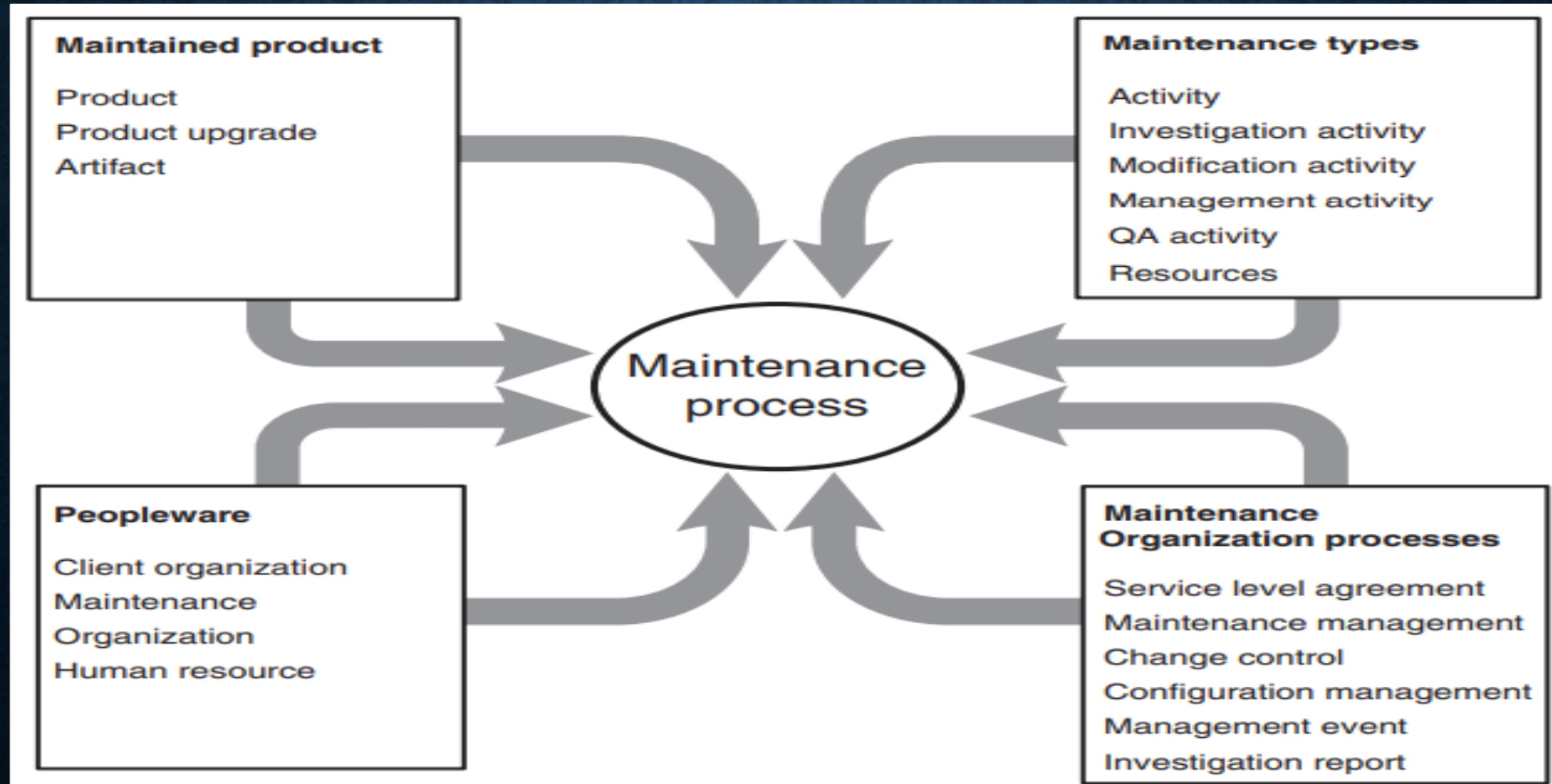
EXAMPLES: SCENARIO TYPE QUESTIONS

- **Example-1:** You are a software development team lead, and your team has recently deployed a new web application for a client. After a few weeks of usage, you start receiving reports from users about performance issues and occasional crashes. As the team lead, you need to determine the appropriate type of maintenance to address these issues.
- **Question:** Based on the scenario described, which type of maintenance strategy would you recommend for addressing the performance issues and occasional crashes in the web application? Provide a brief explanation for your choice.

EXAMPLES: SCENARIO TYPE QUESTIONS

- Example-2: You are the lead developer of a web application used for project management. The application has been in use for a few years, and you have received feedback from users about certain areas that could be improved. For e.g.
 - Substituting components and algorithms: The web application includes a search feature that users have reported to be slow.
 - Modifying naming conventions: During the development of the web application, inconsistent naming conventions were used for variables and functions.
- Question: Based on the scenario described, which type of maintenance strategy would you recommend for addressing mentioned certain areas that could be improved ? Provide a brief explanation for your choice.

CATEGORIES OF MAINTENANCE CONCEPTS



Overview of concept categories affecting software maintenance

MAINTAINED PRODUCT

- The maintained product dimension is characterized by three concepts:
 - **Product:** A product is a coherent collection of several different artifacts. Source code is the key component of a software product.
 - **Product upgrade:** Baseline is an arrangement of related entities that make up a particular software configuration. Any change or upgrade made to a software product relates to a specific baseline.
 - **Artifacts:** A number of different artifacts are used in the design of a software product. One can find the following types of artifacts: textual and graphical documents, component off-the-shelf products, and object code components.
- **The key elements of the maintained product are size, age, application type, composition, and quality.**

MAINTENANCE TYPES

- **Activity:** A number of different broad classes of maintenance activities are performed on software products, including investigation, modification, management, and quality assurance.
- **Investigation activity:** This kind of activities evaluate the impact of making a certain change to the system.
- **Modification activity:** This kind of activities change the system's implementation.
- **Management activity:** This kind of activities relate to the configuration control of the maintained system or the management of the maintenance process.
- **Quality assurance activity:** This kind of activities ensure that modifications performed on a system do not damage the integrity of the system.
- **Resource:** A resource is a **necessary asset whose main role is to help carry out a certain task or project**. A resource can be a person, a team, a tool, finances, and time.

MAINTENANCE ORGANIZATION PROCESSES

- Two different levels of maintenance processes are followed within a maintenance organization:
 - **Individual-level maintenance processes** followed by maintenance personnel to implement a specific change request, and
 - **Organization-level process** followed to manage the change requests from maintenance personnel, users, and customers/clients.

MAINTENANCE ORGANIZATION PROCESSES

Major elements of a maintenance organization are:

- **Service-level agreement (SLA)** : A proposed modification activity is scheduled only after the modification is approved by the board and an Service Level Agreement (SLA) is signed with the client. Service level agreement (SLA) is a contract between the customers and the providers of a maintenance service.
- **Maintenance management**: This process is used to manage the maintenance service, which is not the same as managing individual CRs.
- **Change control**: Evaluation of results of investigations of maintenance events is performed in a process called change control. Based on the evaluation, the organization approves a system change.

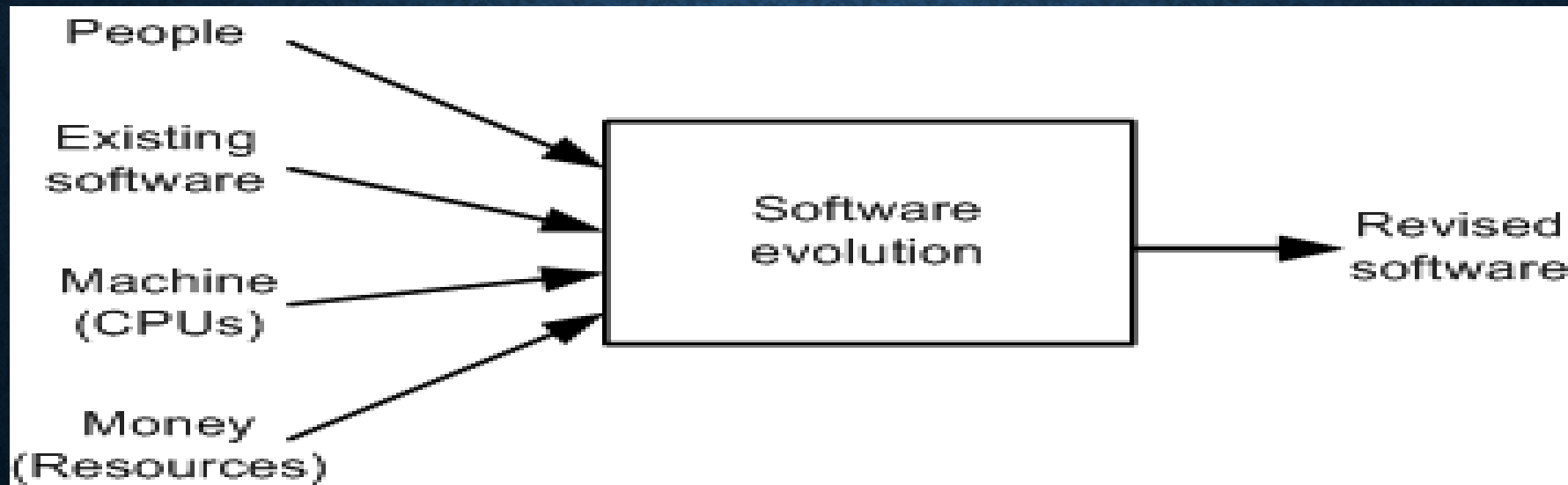
MAINTENANCE ORGANIZATION PROCESSES

- **Configuration management:** A system's integrity is maintained by means of a configuration management process. Integrity of a product is maintained in terms of its modification status and version number.
- **Maintenance event:** This is a problem report or a CR originating from within the maintenance organization or from the customers.
- **Investigation report:** This is the outcome of assessing the cause and impacts of a maintenance event.

PEOPLE WARE

- Maintenance activities cannot ignore the human element, because software production and maintenance are human intensive activities.
- The three people-centric concepts related to maintenance are as follows:
 - **Maintenance organization:** This is the organization that maintains the product(s).
 - **Client organization:** A client organization uses the maintained system.
 - **Human resource:** Human resource includes personnel from the maintenance and client organizations.

EVOLUTION OF SOFTWARE SYSTEMS



Inputs and outputs of software evolution

SUMMARY

- These classifications cover different approaches and motivations behind software maintenance, from fixing issues to proactive upgrades and enhancements, each with distinct activities and tools.
- Software maintenance can be categorized into 12 main types, each type of maintenance plays a vital role in ensuring that software remains functional, adaptable, user-friendly, and resilient over time.