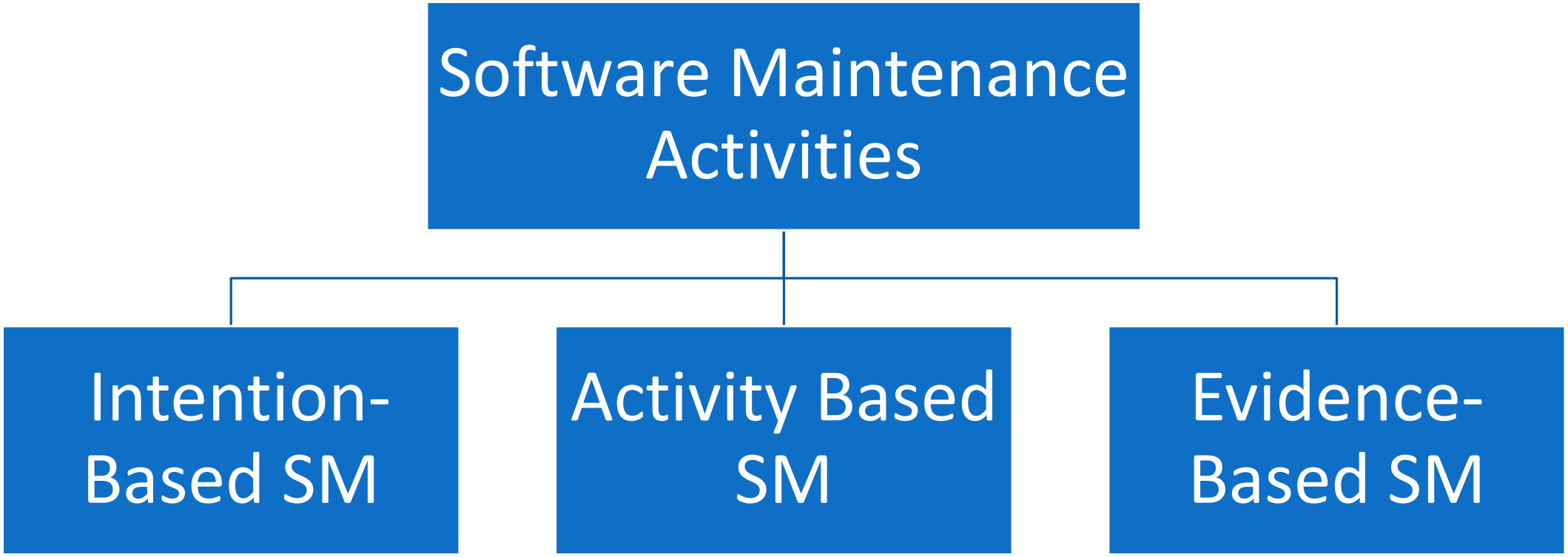
Dr. Lamia Abo Zaid

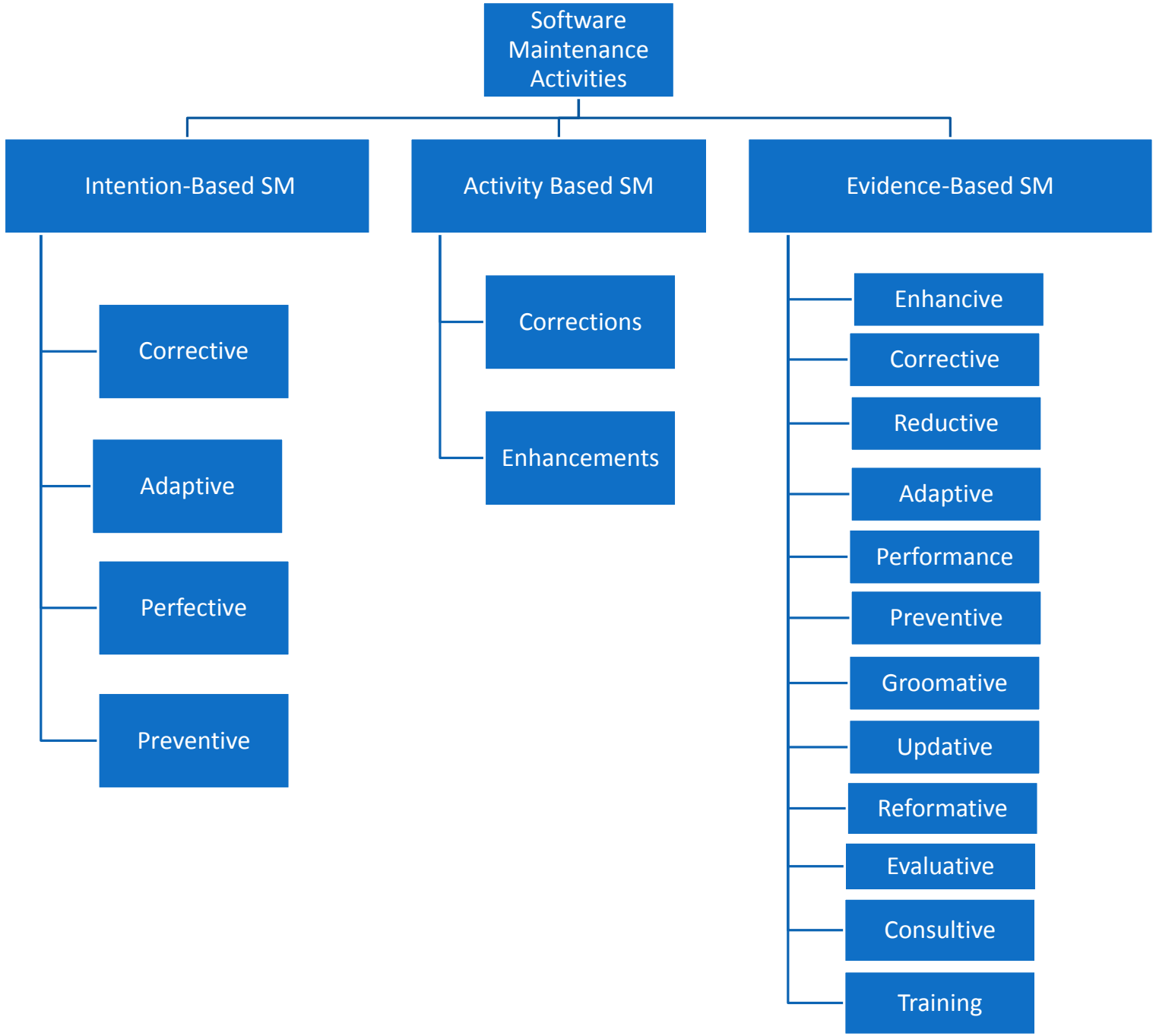
د. لمياء أبوزيد

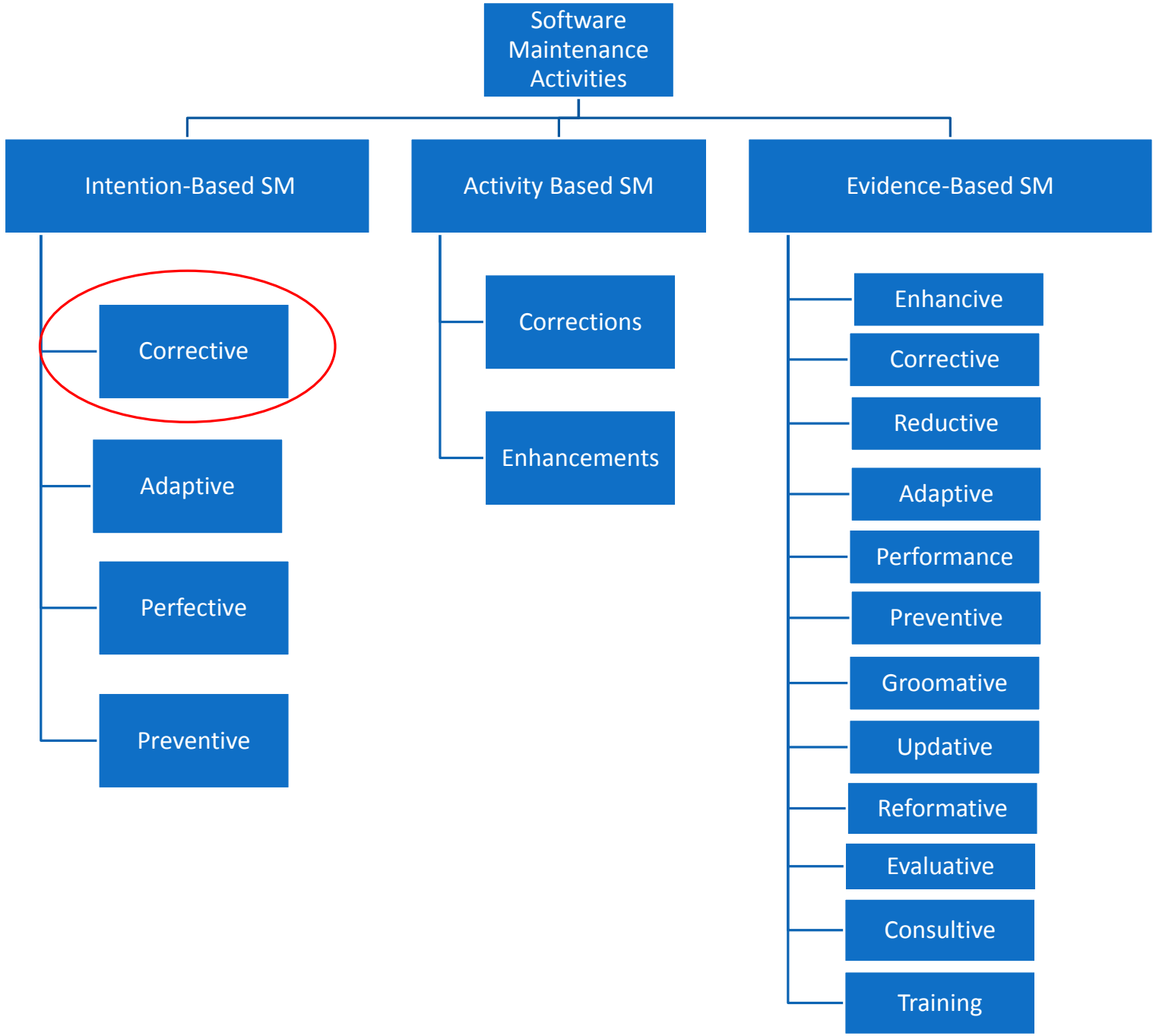
Software Evolution : TOC

1. Introduction to Software Evolution
2. Taxonomy of Software Maintenance and Evolution
3. Evolution and Maintenance Models
4. Reuse and Domain Engineering
5. Program Comprehension
6. Impact Analysis
7. Refactoring
8. Reengineering
9. Legacy Information Systems

Classification of Software Maintenance Activities

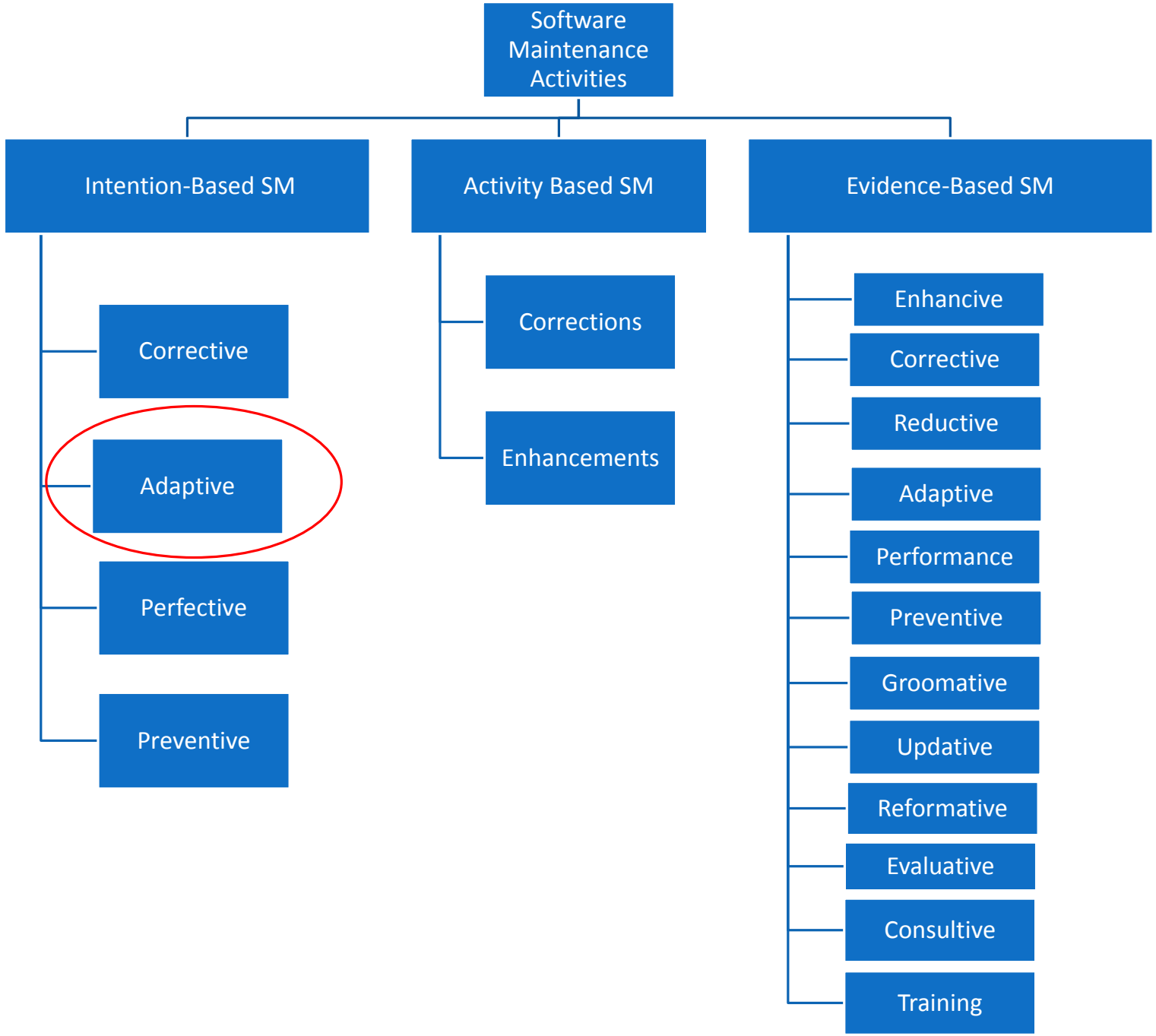



Software Maintenance Activities
Intention-Based SM
Activity Based SM
Evidence-Based SM
Corrective
Adaptive
Perfective
Preventive
Corrections
Enhancements
Enhancive
Corrective
Reductive
Adaptive
Performance
Preventive
Groomative
Updative
Reformative
Evaluative
Consultive
Training


Software Maintenance Activities
Intention-Based SM
Activity Based SM
Evidence-Based SM
Corrective
Adaptive
Perfective
Preventive
Corrections
Enhancements
Enhancive
Corrective
Reductive
Adaptive
Performance
Preventive
Groomative
Updative
Reformative
Evaluative
Consultive
Training

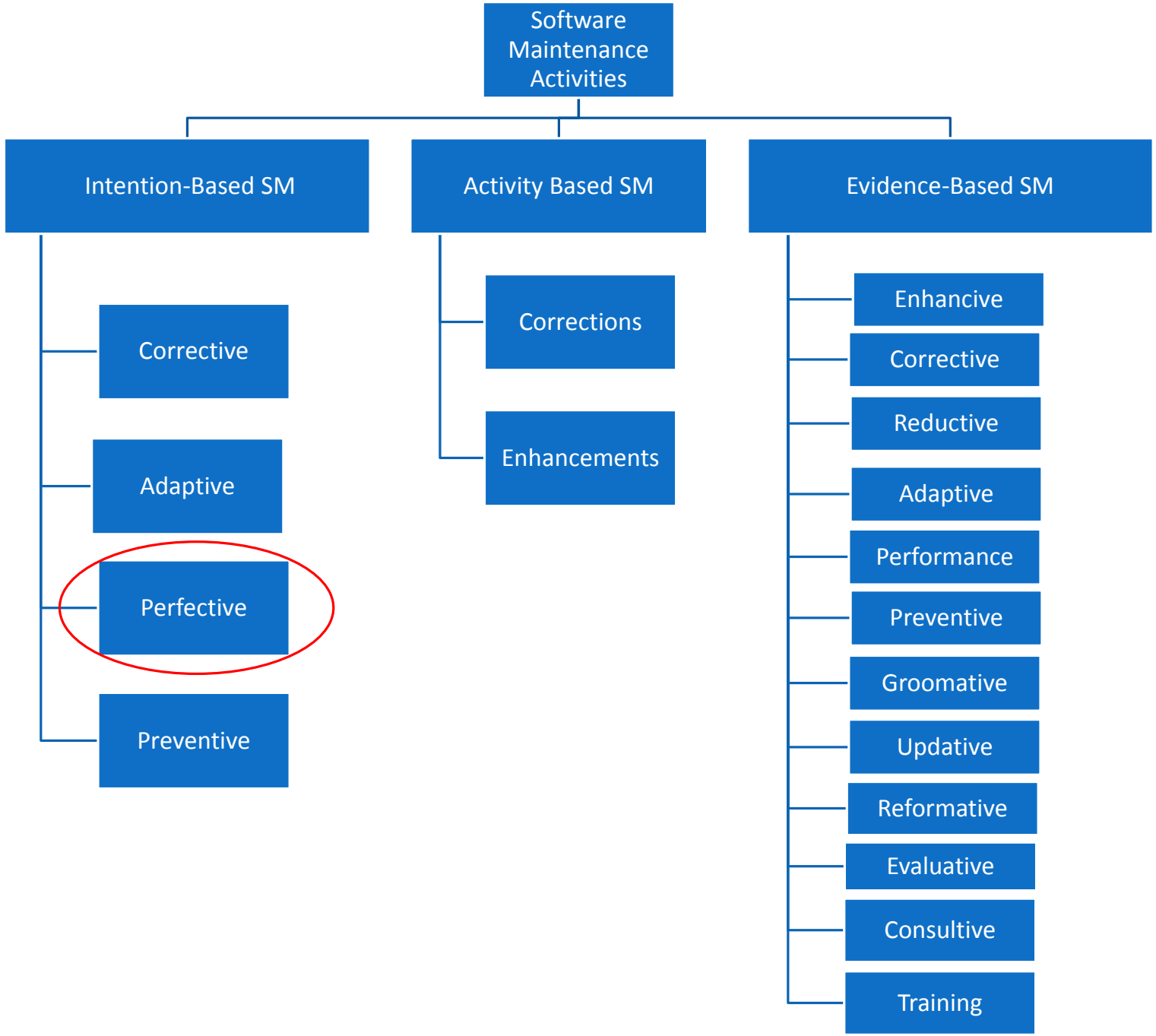
Intention-based Classification of Software Maintenance - Corrective Maintenance

- **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 → processing failure.
  - a program that aborts or produces incorrect results → processing failure.
  - A program not being able to meet real-time requirements → performance failure.
- The process of corrective maintenance includes isolation and correction of defective elements in the software.
- Corrective maintenance is a reactive process, which means that corrective maintenance is performed after detecting defects with the system.


Software Maintenance Activities
Intention-Based SM
Activity Based SM
Evidence-Based SM
Corrective
Adaptive
Perfective
Preventive
Corrections
Enhancements
Enhancive
Corrective
Reductive
Adaptive
Performance
Preventive
Groomative
Updative
Reformative
Evaluative
Consultive
Training

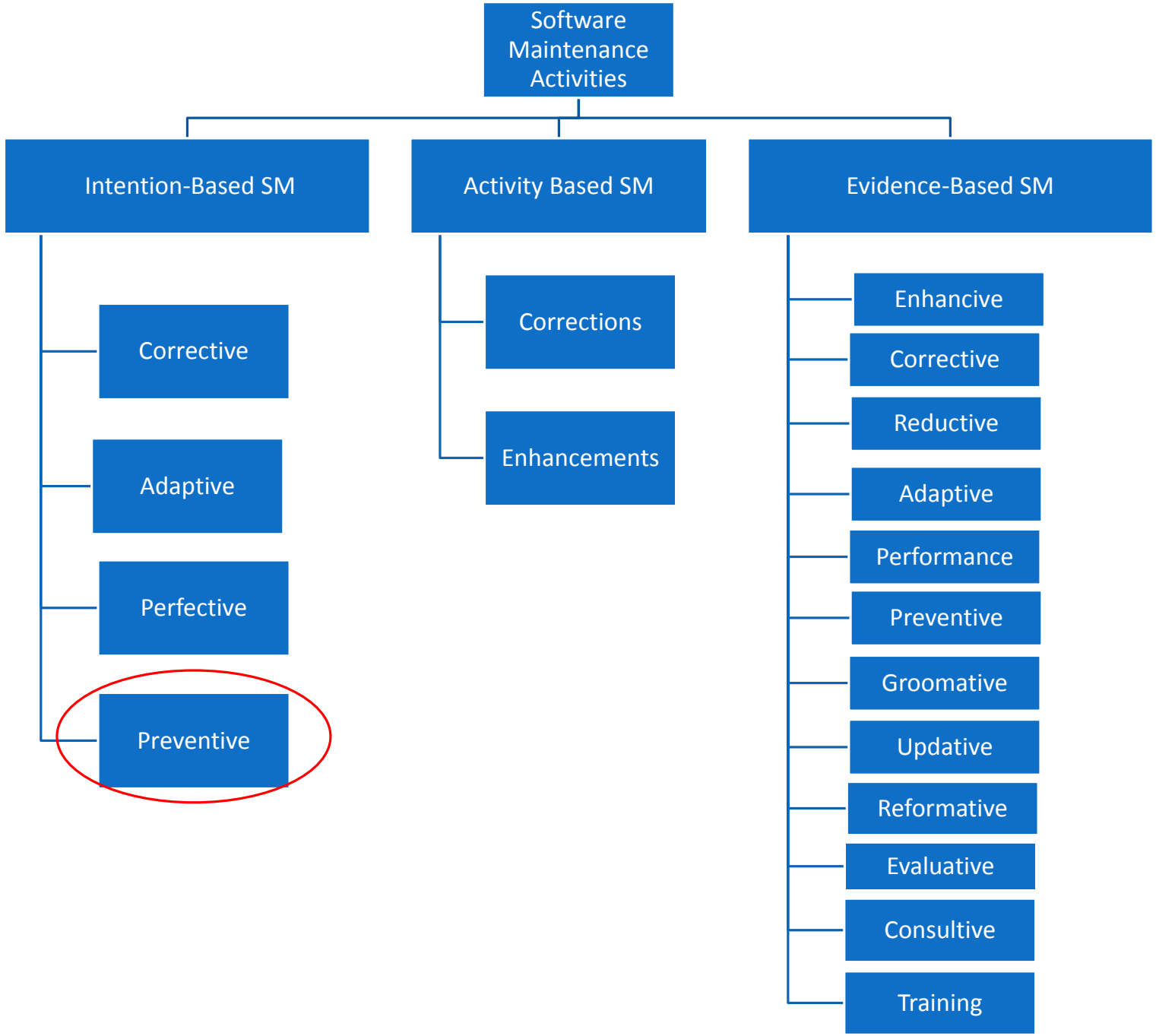
Intention-based Classification of Software Maintenance - Adaptive Maintenance

- **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
  - changing the system to be compatible with other applications (OS or Database)


Software Maintenance Activities
Intention-Based SM
Activity Based SM
Evidence-Based SM
Corrective
Adaptive
Perfective
Preventive
Corrections
Enhancements
Enhancive
Corrective
Reductive
Adaptive
Performance
Preventive
Groomative
Updative
Reformative
Evaluative
Consultive
Training

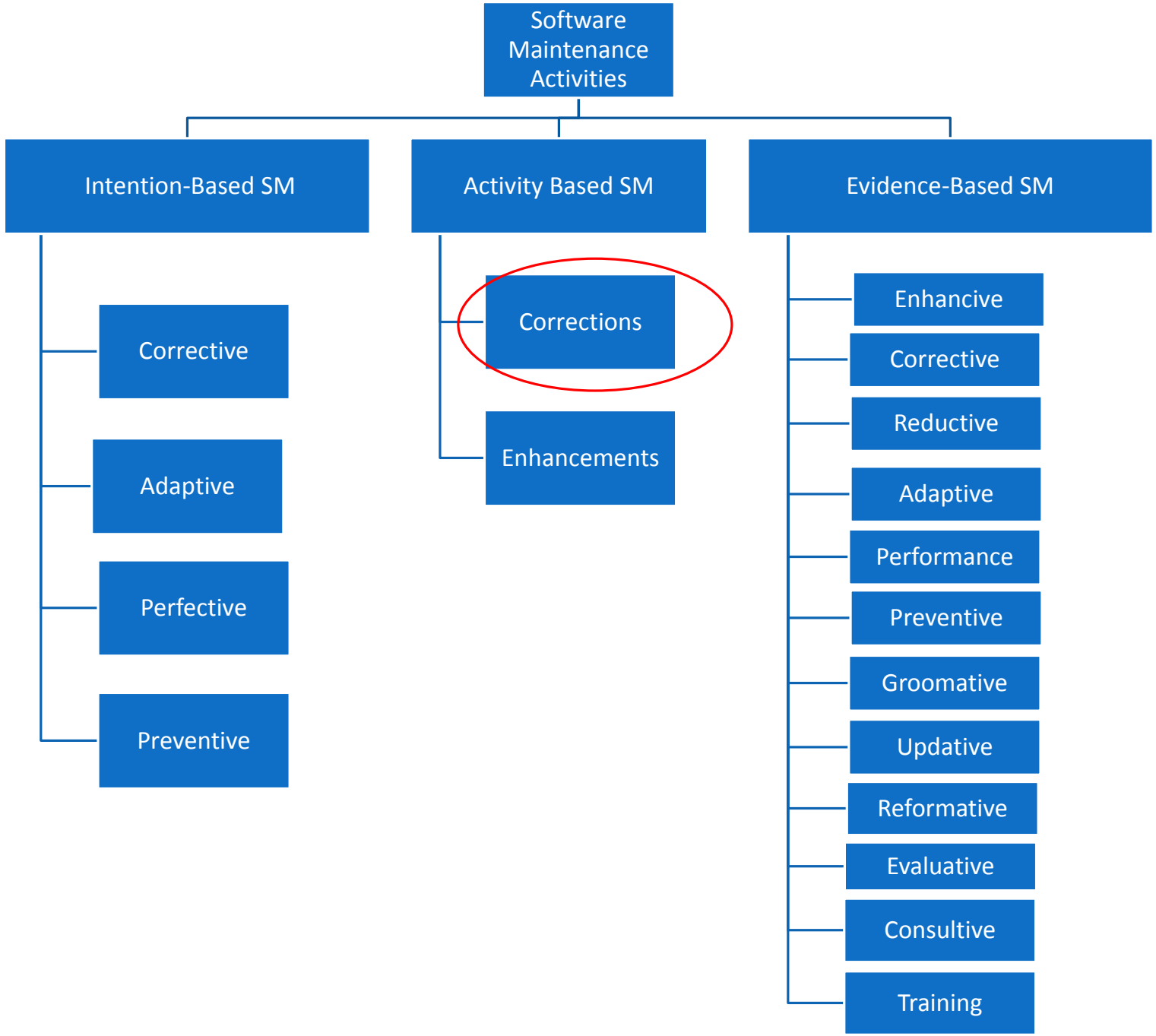
Intention-based Classification of Software Maintenance - Perfective Maintenance

- **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".


Software Maintenance Activities
Intention-Based SM
Activity Based SM
Evidence-Based SM
Corrective
Adaptive
Perfective
Preventive
Corrections
Enhancements
Enhancive
Corrective
Reductive
Adaptive
Performance
Preventive
Groomative
Updative
Reformative
Evaluative
Consultive
Training

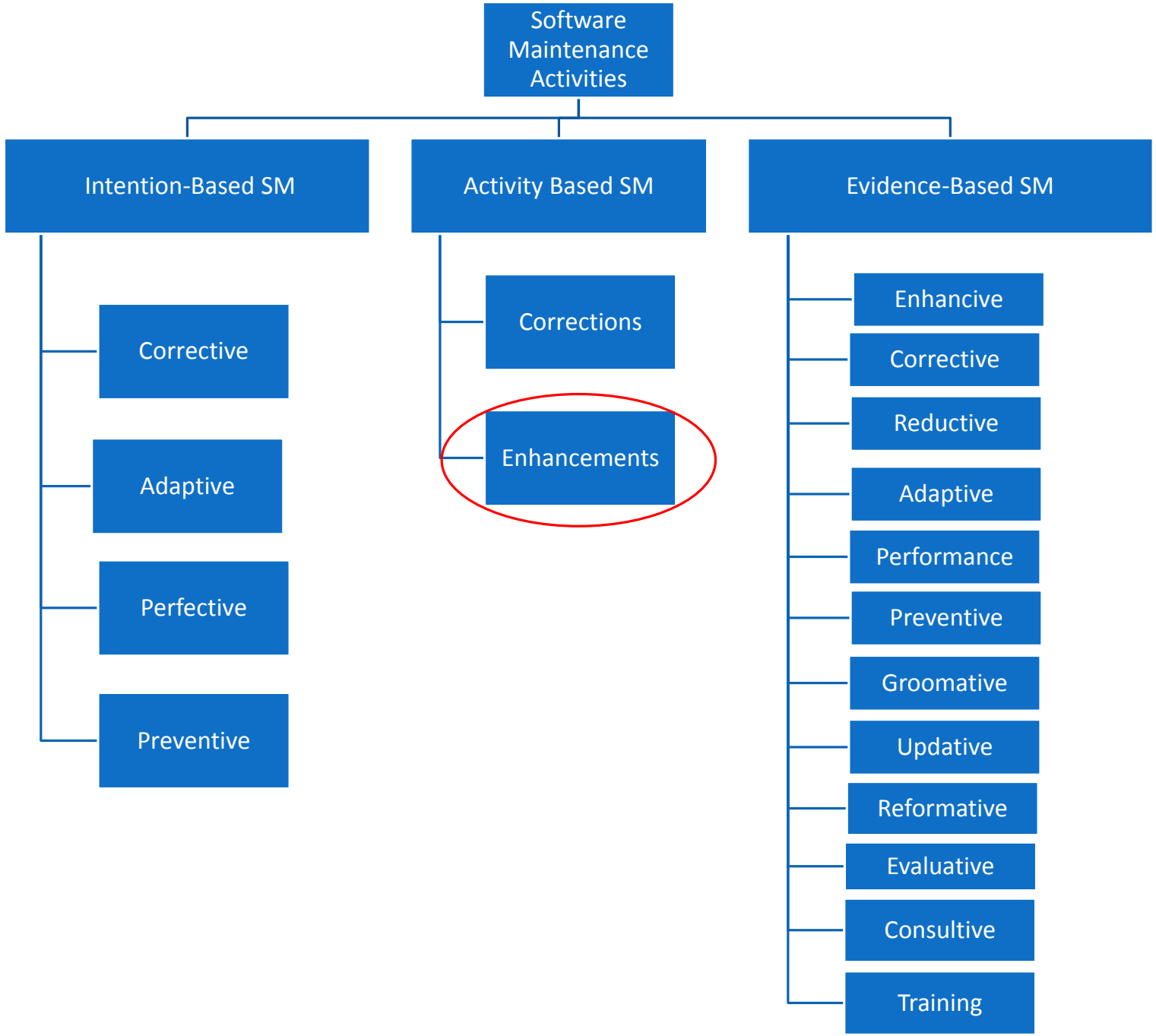
Intention-based Classification of Software Maintenance - Preventive Maintenance

- **Preventive maintenance:** The purpose of preventive maintenance is to prevent problems from occurring by modifying software products. to eliminate hazard or reduce its associated risk to an acceptable level.
  - a hazard is a state of a system or a physical situation which, when combined with certain environment conditions, could lead to an accident.
- Preventive maintenance is identifying future risks and unknown problems, and taking actions so that those problems do not occur. It involves occasionally terminating an application or a system, cleaning its internal state, and restarting it.
- Preventive maintenance is very often performed on safety critical and high available software systems (eg. Control software for Patriot Missile, control software for telecommunication services).
- The concept of "software rejuvenation" is a preventive maintenance measure to prevent, or at least postpone, the occurrences of failures (crash) resulting from continuously running the software system.


Software Maintenance Activities
Intention-Based SM
Activity Based SM
Evidence-Based SM
Corrective
Adaptive
Perfective
Preventive
Corrections
Enhancements
Enhancive
Corrective
Reductive
Adaptive
Performance
Preventive
Groomative
Updative
Reformative
Evaluative
Consultive
Training

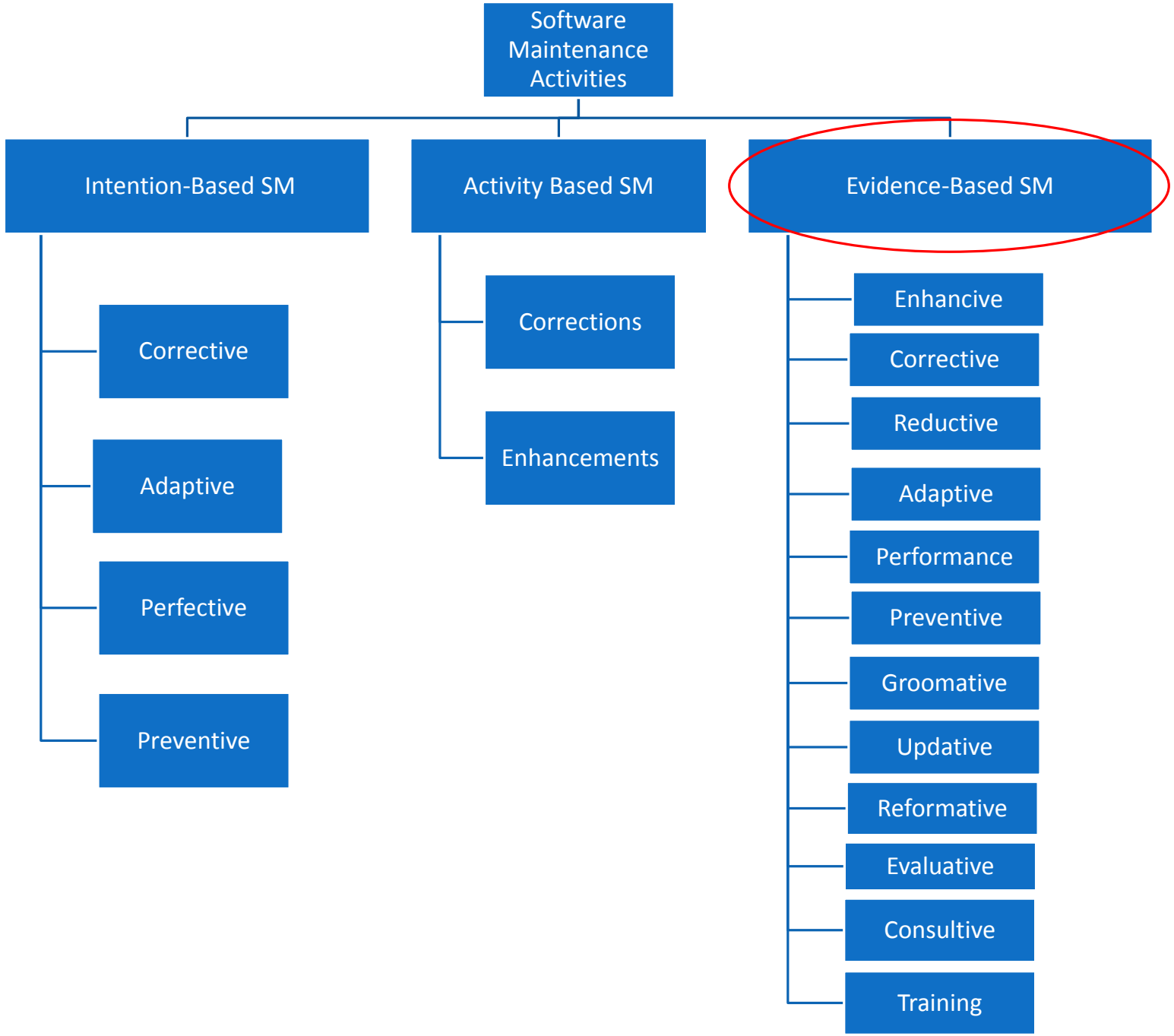
Activity-based Classification of Software Maintenance -Corrections

- Corrections: Activities in this category are designed to fix defects in the system, where a defect is a discrepancy between the expected behaviour and the actual behaviour of the system.
- e.g. wrong calculation of taxes in a payroll component of the system
- e.g. usage of wrong metrics for calculation


Software Maintenance Activities
Intention-Based SM
Activity Based SM
Evidence-Based SM
Corrective
Adaptive
Perfective
Preventive
Corrections
Enhancements
Enhancive
Corrective
Reductive
Adaptive
Performance
Preventive
Groomative
Updative
Reformative
Evaluative
Consultive
Training

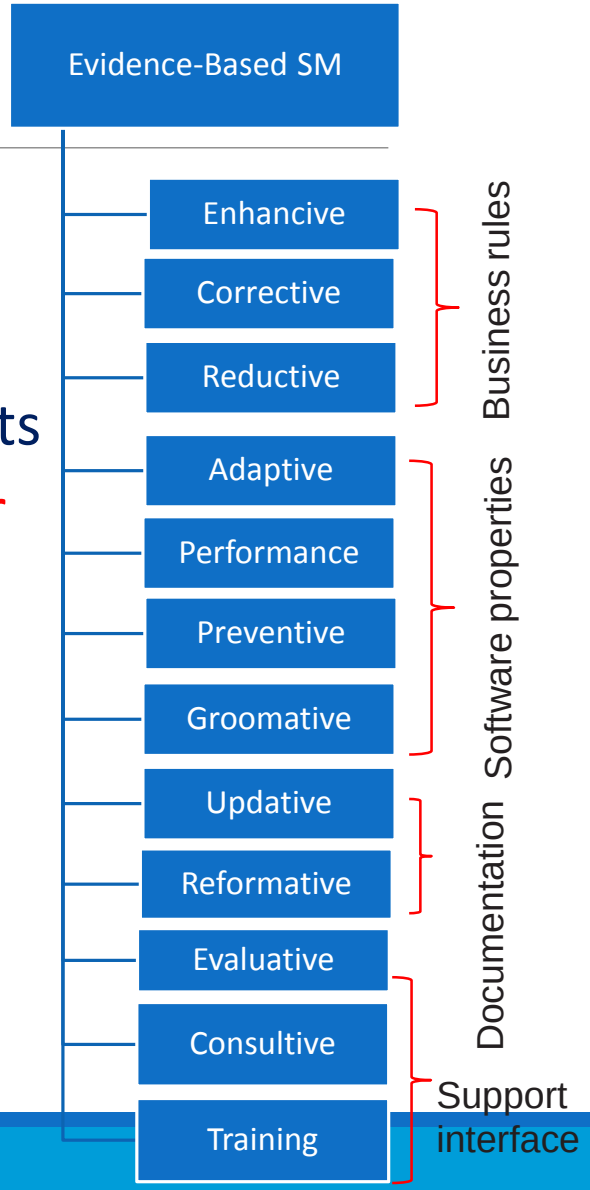
Activity-based Classification of Software Maintenance - Enhancements

- Enhancements Maintenance activities are designed to effect changes to the system.
- It 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. => Refactoring/ Reengineering


Software Maintenance Activities
Intention-Based SM
Activity Based SM
Evidence-Based SM
Corrective
Adaptive
Perfective
Preventive
Corrections
Enhancements
Enhancive
Corrective
Reductive
Adaptive
Performance
Preventive
Groomative
Updative
Reformative
Evaluative
Consultive
Training

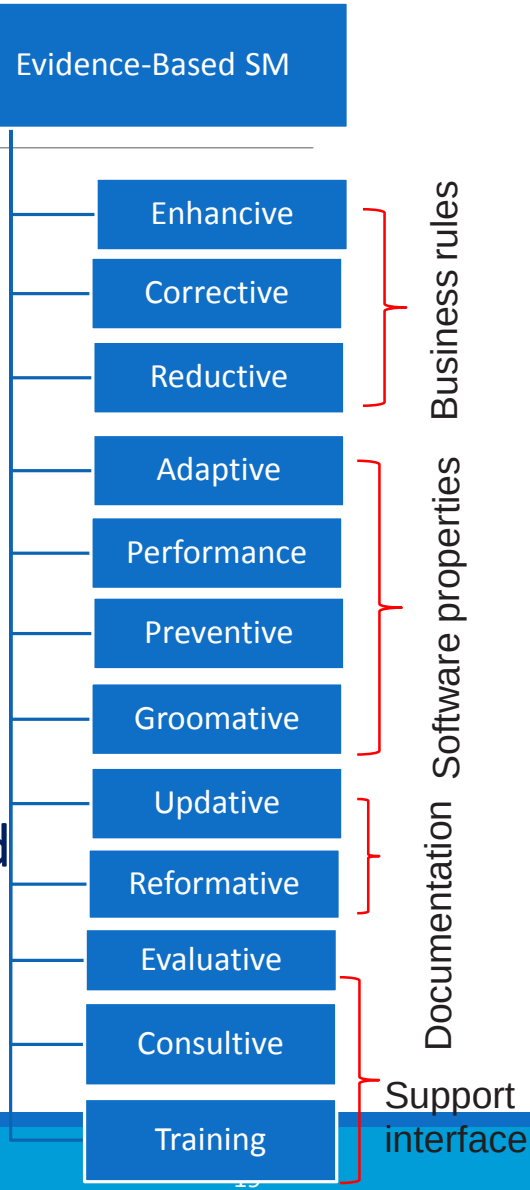
Evidence-based Classification of Software Maintenance

- Twelve mutually exclusive types of maintenance activities were grouped into four clusters.
- 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, these are:
  - the whole software
  - the external documentation
  - the properties of the program code
  - the system functionality experienced by the customer



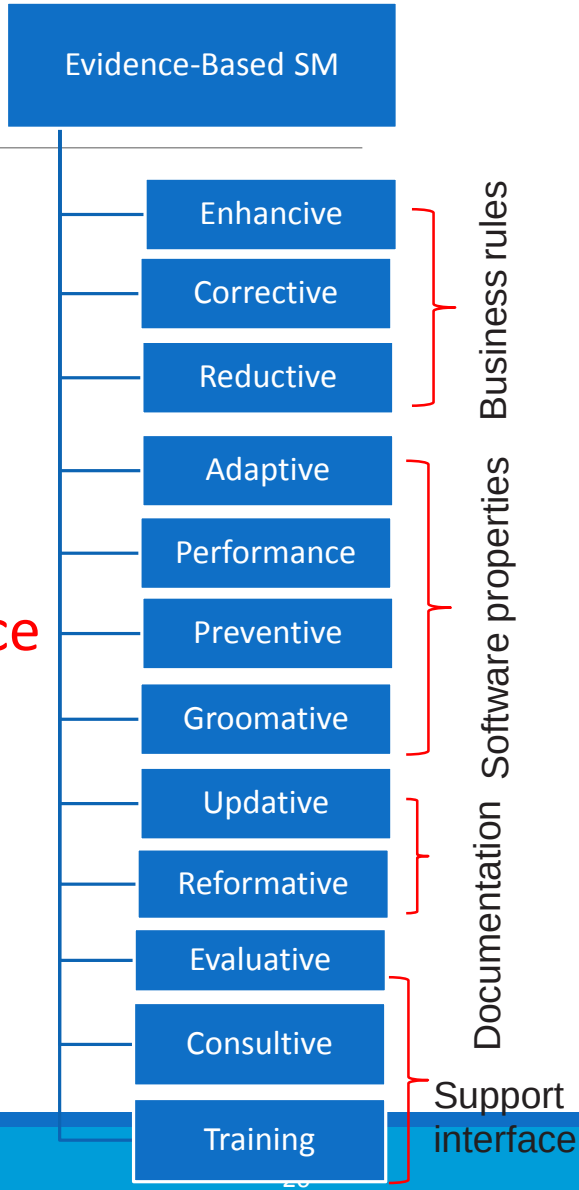
Evidence-based Classification of Software Maintenance

- **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.



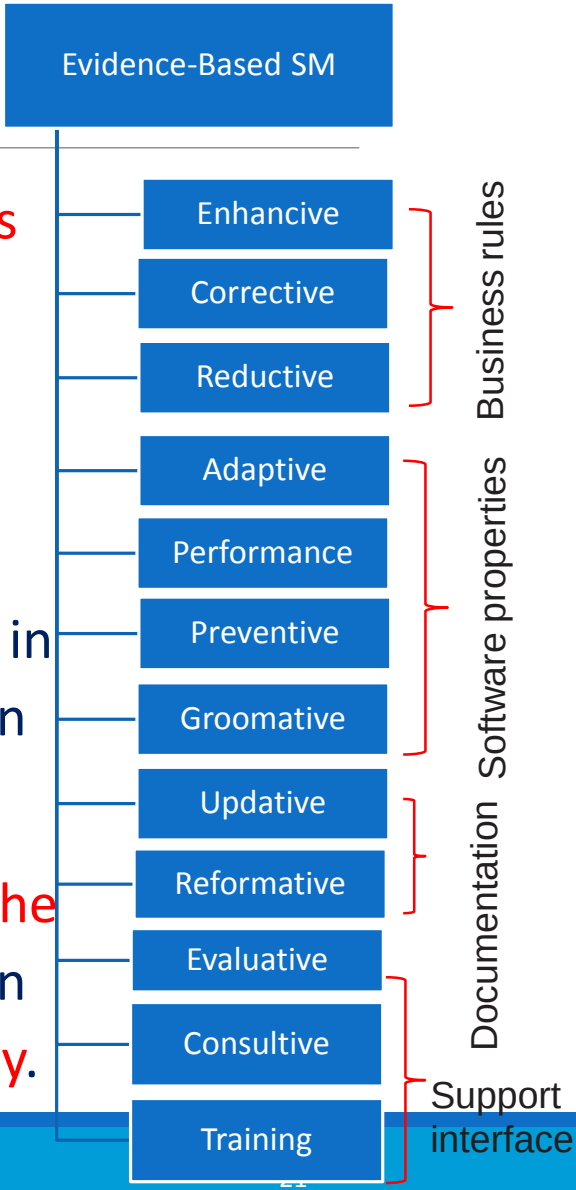
Evidence-based Classification of Software Maintenance

- **Adaptive**: Ordinary activities in this type port the software to a different execution platform, and increase the utilization of COTS 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.



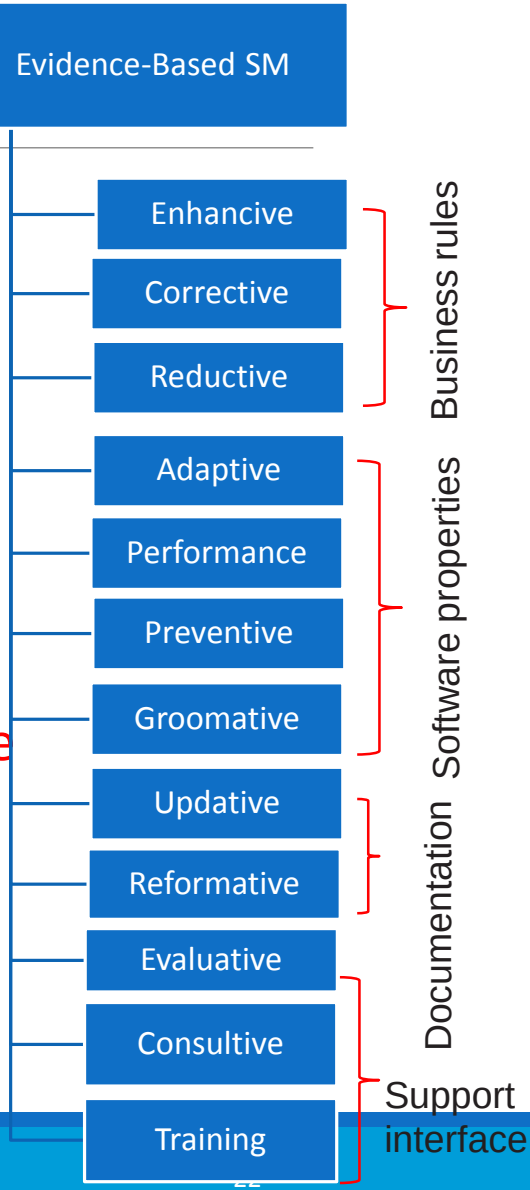
Evidence-based Classification of Software Maintenance

- **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.
- **Updative**: 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 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.



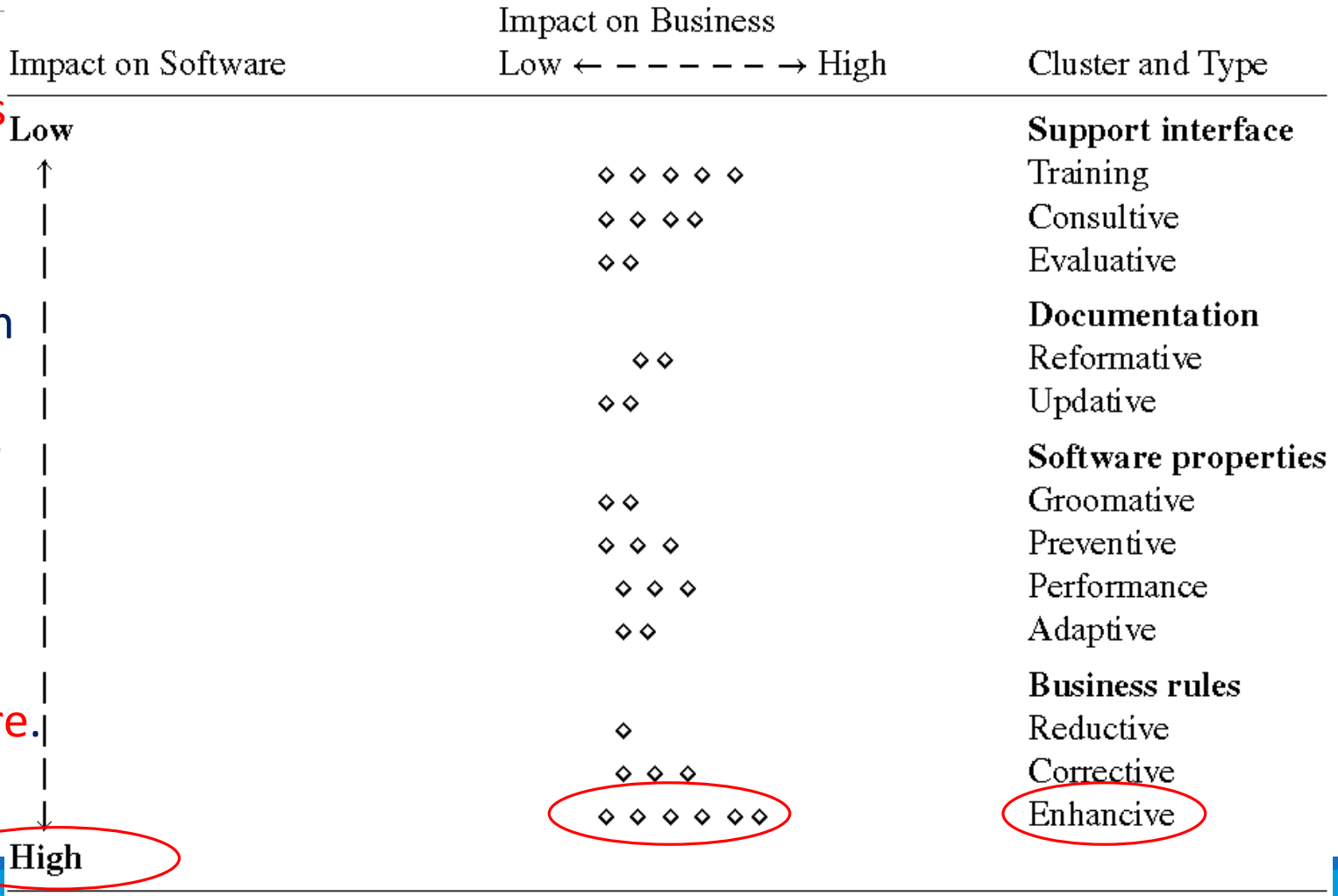
Evidence-based Classification of Software Maintenance

- **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 – Impact of Maintenance

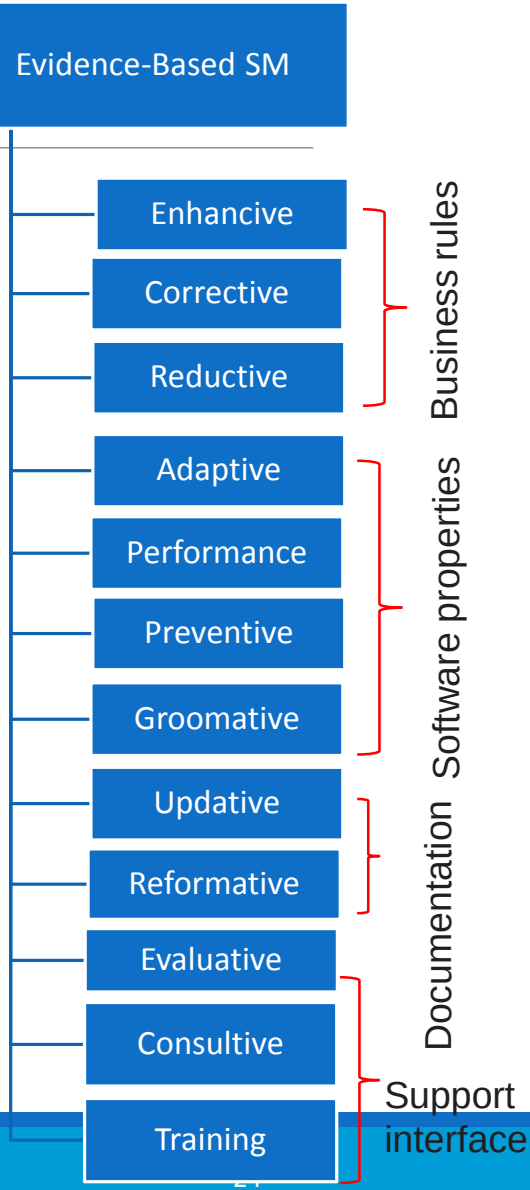
- The first dimension is the customer's ability to perform its business function while continuing to use the system.
  - E.g. if the software is enhanced with new functionalities, then the customer is more likely to be able to achieve its business goals than modifications on non-code documentation.
- The second dimension is the software. This is arranged from top to bottom.



Source: From Reference 15. © 2001 John Wiley & Sons.

Evidence-based Classification of Software Maintenance - Decision Criteria

- The classification is based on modifications deliberately done, modifications observed, modifications manifested, and modifications detected. With respect to :
  - A—the whole software system.
  - B—the program code.
  - C—the functionalities experienced by customers.
  - D—the external documentation
- Activities are classified into different types by applying a two-step decision process:
  - First, apply criteria-based decisions to make the clusters of types.
  - Next, apply the type decisions to identify one type within the cluster.



Notes:
Types is read from left to right, bottom to top.
Questions have been listed in Table 2.2.
Italics show the type name when the type decision at the left of it is 'Yes'
For "cc," read "change to code"
* indicates the default type in the cluster.

A—the whole software system.
B—the program code.
C—the functionalities experienced by customers.
D—the external documentation

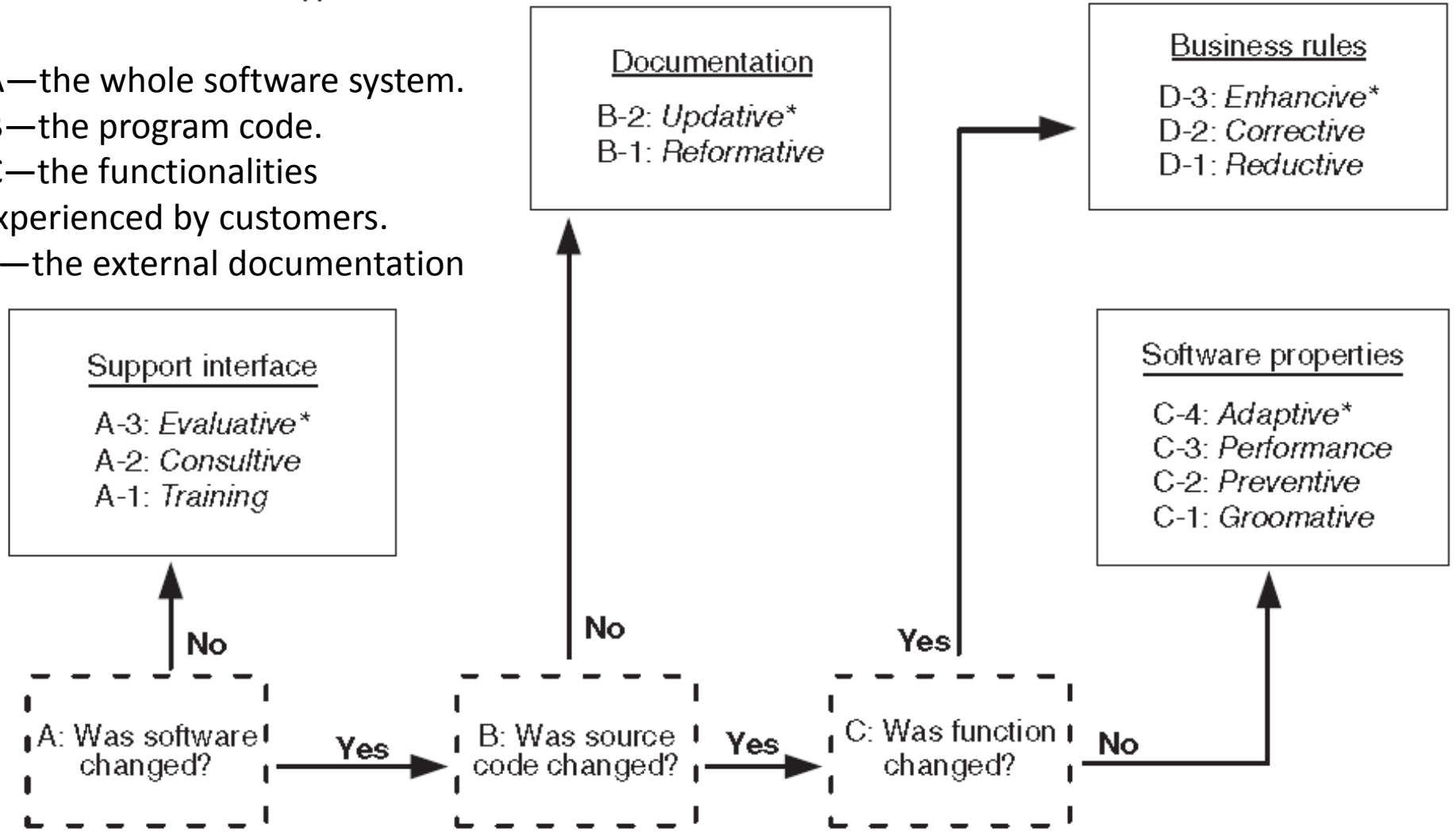
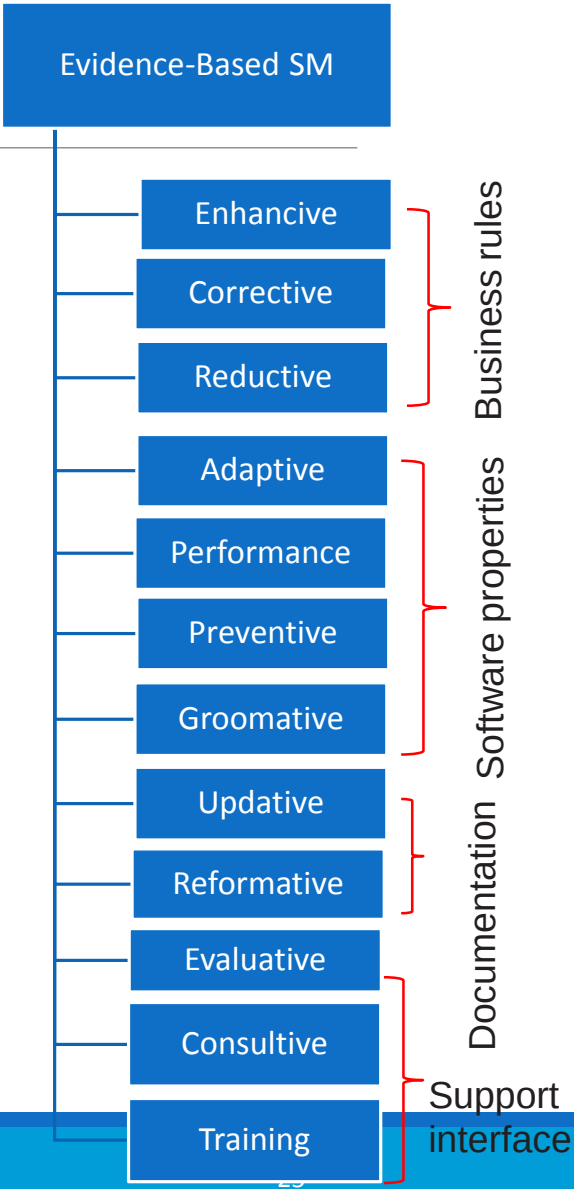
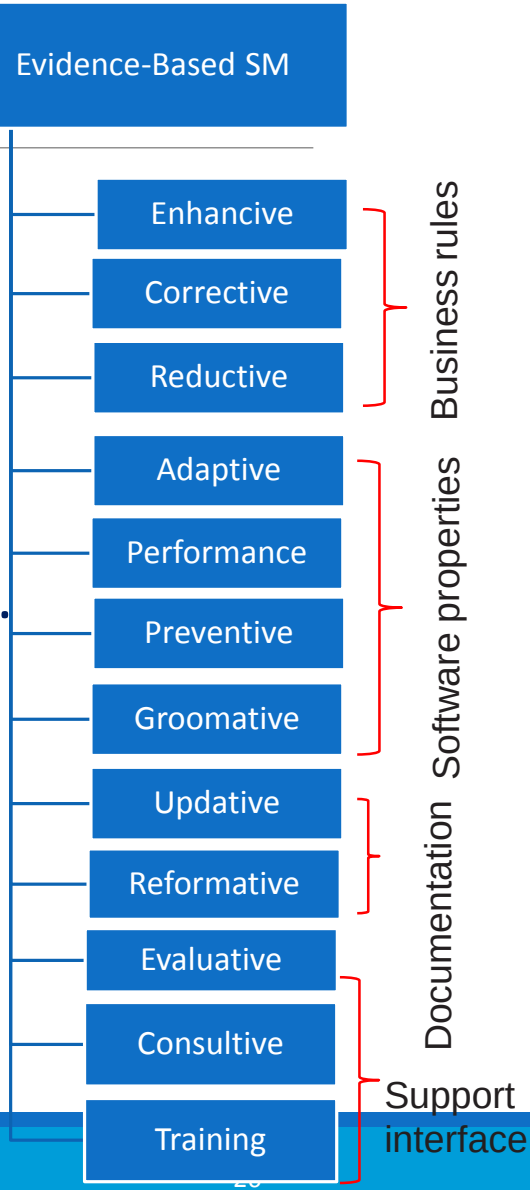


FIGURE 2.2 Decision tree types. From Reference 15. © 2001 John Wiley & Sons



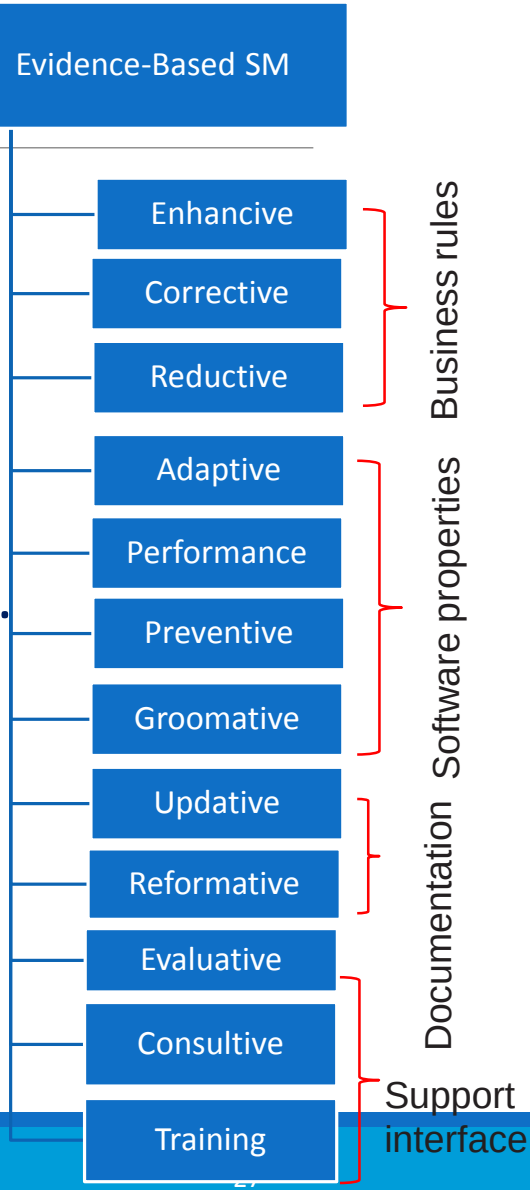
Evidence-based Classification of Software Maintenance - Decision Criteria

- A maintenance engineer, after analyzing all the documentation along with the program code, modified the program code for one component without modifying other documentation, built the rewritten component, executed the regression test suite, checked it into the version control, and embedded it into the production system. The only consequence the customer observed was improved latency.
- Question: Identify the type of software maintenance performed by the engineer



Evidence-based Classification of Software Maintenance - Decision Criteria

- A maintenance engineer, after analyzing all the documentation along with the program code, modified the program code for one component without modifying other documentation, built the rewritten component, executed the regression test suite, checked it into the version control, and embedded it into the production system. The only consequence the customer observed was improved latency.
- Question: Identify the type of software maintenance performed by the engineer



Evidence-based Classification of Software Maintenance - Decision Criteria

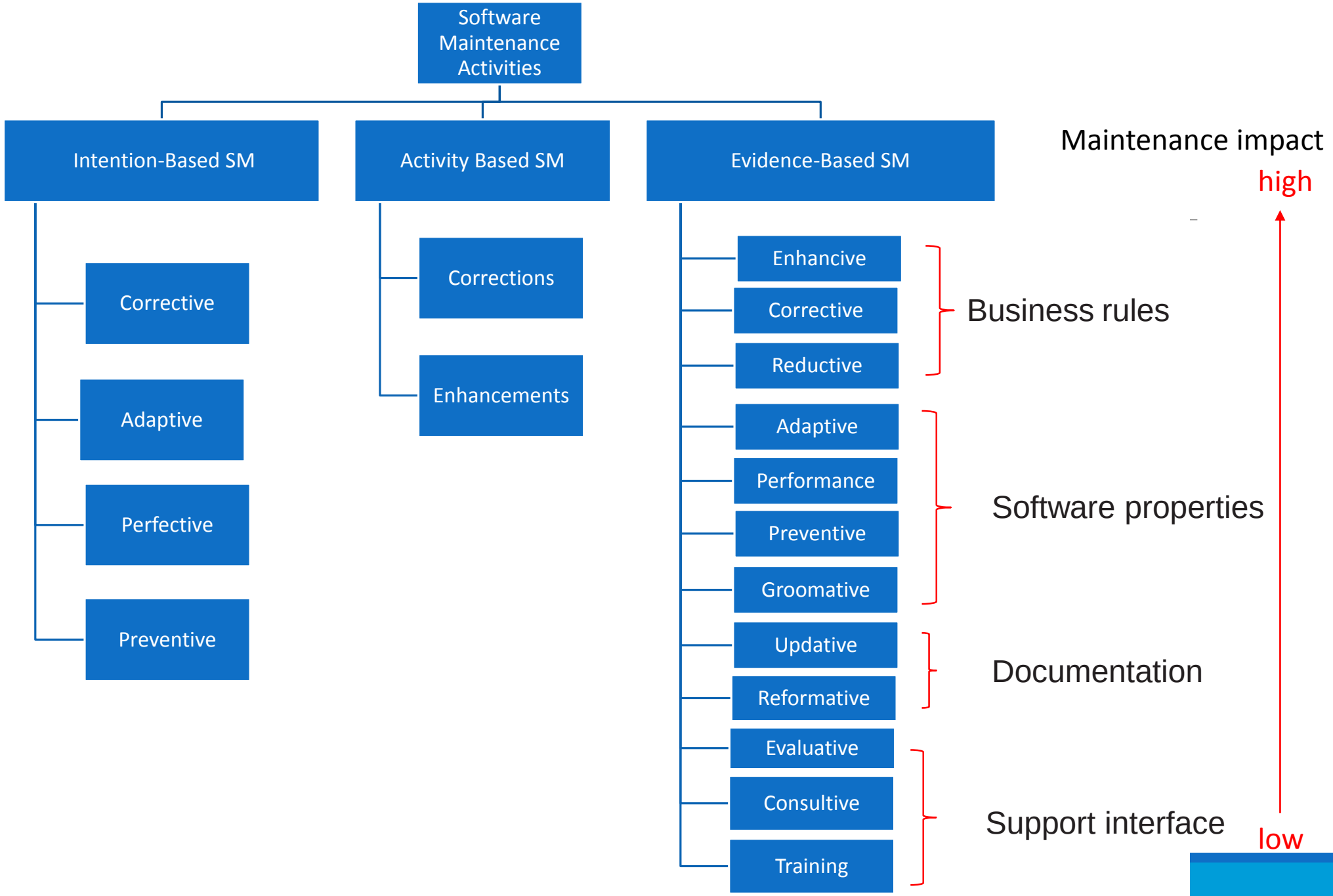
A—the whole software system.
B—the program code.
C—the functionalities experienced by customers.
D—the external documentation

Criteria	Type decision question	Type
A-1	To train the stakeholders, did the activities utilize the software as subject?	Training
A-2	As a basis for consultation, did the activities employ the software?	Consultive
A-3	Did the activities evaluate the software?	Evaluative
B-1	To meet stakeholder needs, did the activities modify the non-code documentation?	Reformative
B-2	To conform to implementation, did the activities modify the non-code documentation?	Updative
C-1	Was maintainability or security changed by the activities?	Groomative
C-2	Did the activities constrain the scope of future maintenance activities?	Preventive
C-3	Were performance properties or characteristics modified by the activities?	Performance
C-4	Were different technology or resources used by the activities?	Adaptive
D-1	Did the activities constrain, reduce, or remove some functionalities experienced by the customer?	Reductive
D-2	Did the activities fix bugs in customer-experienced functionality?	Corrective
D-3	Did the activities substitute, add to, or expand some functionalities experienced by the customers?	Enhancive

Table 2.3: Summary of evidence-based types of software maintenance

Evidence-based Classification of Software Maintenance - Decision Criteria

- Sometimes, an objective evidence may be found to be ambiguous. In that case, clusters have their designated default types for use.
- The overall default type is evaluative, if there are ambiguities in an activity.


Software Maintenance Activities
Intention-Based SM
Activity Based SM
Evidence-Based SM
Corrective
Adaptive
Perfective
Preventive
Corrections
Enhancements
Enhancive
Corrective
Reductive
Adaptive
Performance
Preventive
Groomative
Updative
Reformative
Evaluative
Consultive
Training
Business rules
Software properties
Documentation
Support interface
Maintenance impact
high
low

Questions

?

Readings

- Book Chapter 2
  - Section 2.1, 2.2