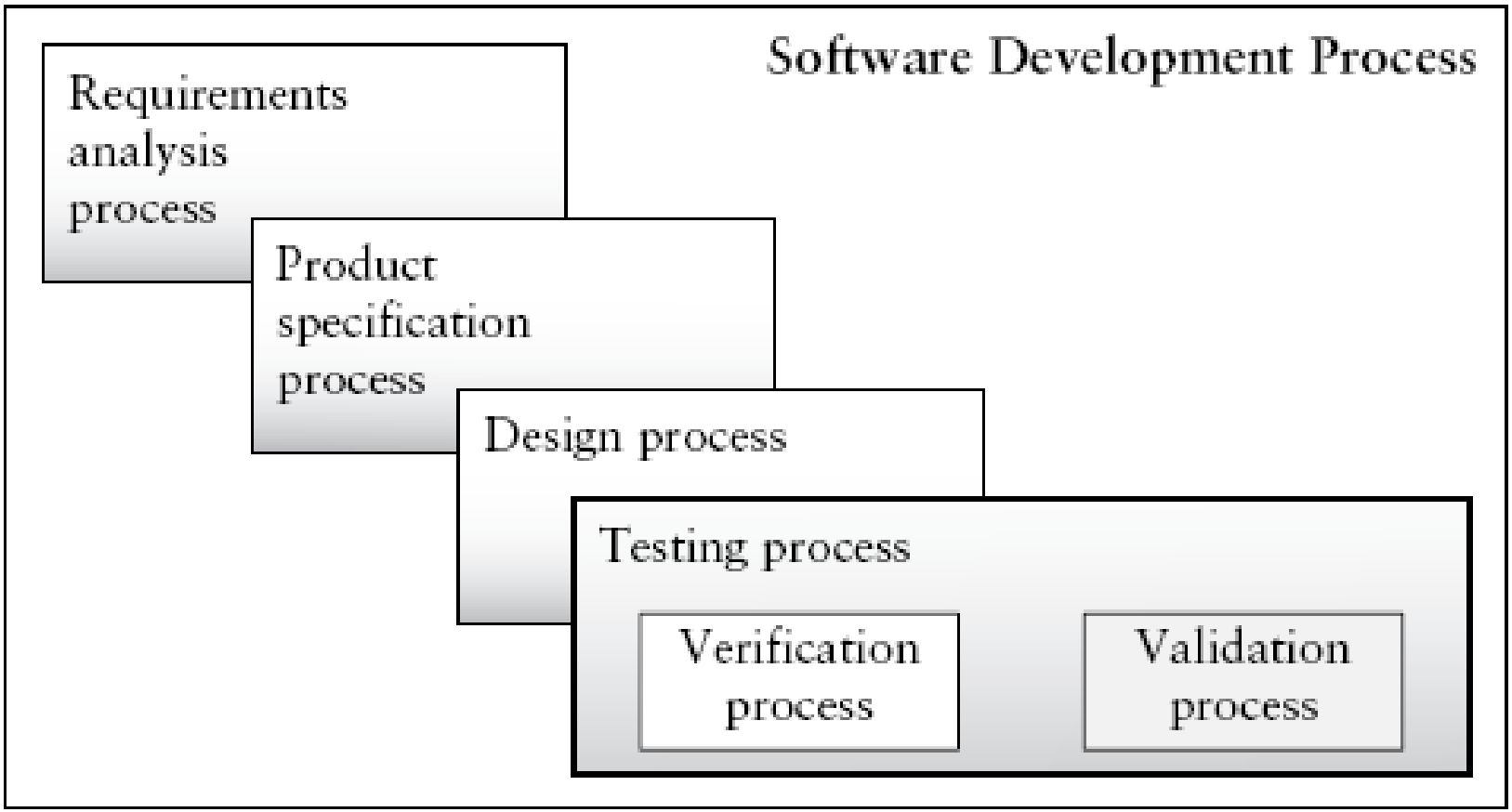
TESTING AS A PROCESS



SOFTWARE TESTING PRINCIPLES

Principle-1: Testing is the process of exercising a software component using a selected set of tests cases, with the internet.

- Revealing defects, and Evaluating quality.
- Software engineers have made great progress in developing methods to prevent and eliminate defects. However, defects do occur, and they have a negative impact on a software quality. This principle supports testing as an execution-based activity to detect defects.
- The term **defect** as used in this and in subsequent principle represents any deviations in the software that have negative impact on its functionality, performance, reliability, security and other of its specified quality attributes.

Principle-2:When the test objectives is to detect defects, then a good test case is one that has a high probability of revealing a yet undetected defects.

- The goal for the test is to prove / disprove the hypothesis that is, determine if the specific defect is present / absent.
- A tester can justify the expenditure of the resources by careful test design so that principle two is supported.

Principle-3:Test result should be inspected meticulously.

- Tester need to carefully inspect and interpret test results. Several erroneous and costly scenarios may occur if care is not taken.

Example:

- A failure may be overloaded, and the test may be granted a pass status when in reality the software has failed the test. Testing may continue based on erroneous test result. The defect may be revealed at some later stage of testing, but in that case it may be make costly and difficult to locate and repair.

Principle-4: A test case must contain the expected output or result.

- The test case is of no value unless there is an explicit statement of the expected outputs or results.

Example:

A specific variable value must be observed or a certain panel button that must light up.

Principle-5: Test cases should be developed for both valid and invalid input conditions.

- The tester must not assume that the software under test will always be provided with valid inputs.
- Inputs may be incorrect for several reasons.

Example:

Software users may have misunderstandings, or lack information about the nature of the inputs. They often make typographical errors even when compute / correct information are available. Device may also provide invalid inputs due to erroneous conditions and malfunctions.

Principle-6: The probability of the existence of additional defects in a software component is proportional to the number of defects already defected in that component.

Example:

- If there are two components A and B and testers have found 20 defects in A and 3 defects in B, then the probability of the existence of additional defects in A is higher than B.

Principle-7: Testing should be carried out by a group that is independent of the development group.

- Tester must realize that
  - Developers have a great deal of pride in their work and
  - On practical level it may be difficult for them to conceptualize where defects could be found.

Principle-8: Tests must be repeatable and reusable

- This principle calls for experiments in the testing domain to require recording of the exact condition of the test, any special events that occurred, equipment used, and a careful accounting of the results.
- This information invaluable to the developers when the code is returned for debugging so that they can duplicate test conditions.

Principle-9: Testing should be planned.

- Test plan should be developed for each level of testing, and objective for each level should be described in the associated plan.

- The objectives should be stated as quantitatively as possible plan, with their precisely specified objectives.

Principle-10: Testing activities should be integrated into the software life cycle.

- It is no longer feasible to postpone testing activities until after the code has been written.
- Test planning activities into the software lifecycle starting as early as in the requirements analysis phases, and continue on throughout the software lifecycle in parallel with development activities.

Principle-11: Testing is a creative and challenging task.

- Difficult and challenges for the tester includes,
- A tester needs to have comprehensive knowledge of the software engineering discipline.
- A tester needs to have knowledge from both experience and education as to how software is specified, designed and developed.
- A tester needs to be able to manage many details.
- A tester needs to have knowledge of fault type and where faults of a certain type night occur in code constructs.
- A tester needs a reason like scientist and propose hypotheses that related to presence of specific type of defects.
- A tester needs to design and record test procedure for running the tests.
- A tester to plan for testing and allocate the proper resources.
- A tester need to execute the tests and is responsible for recording results.
- A tester need to learn to use tools and keeps abreast of the newest test tool advance.

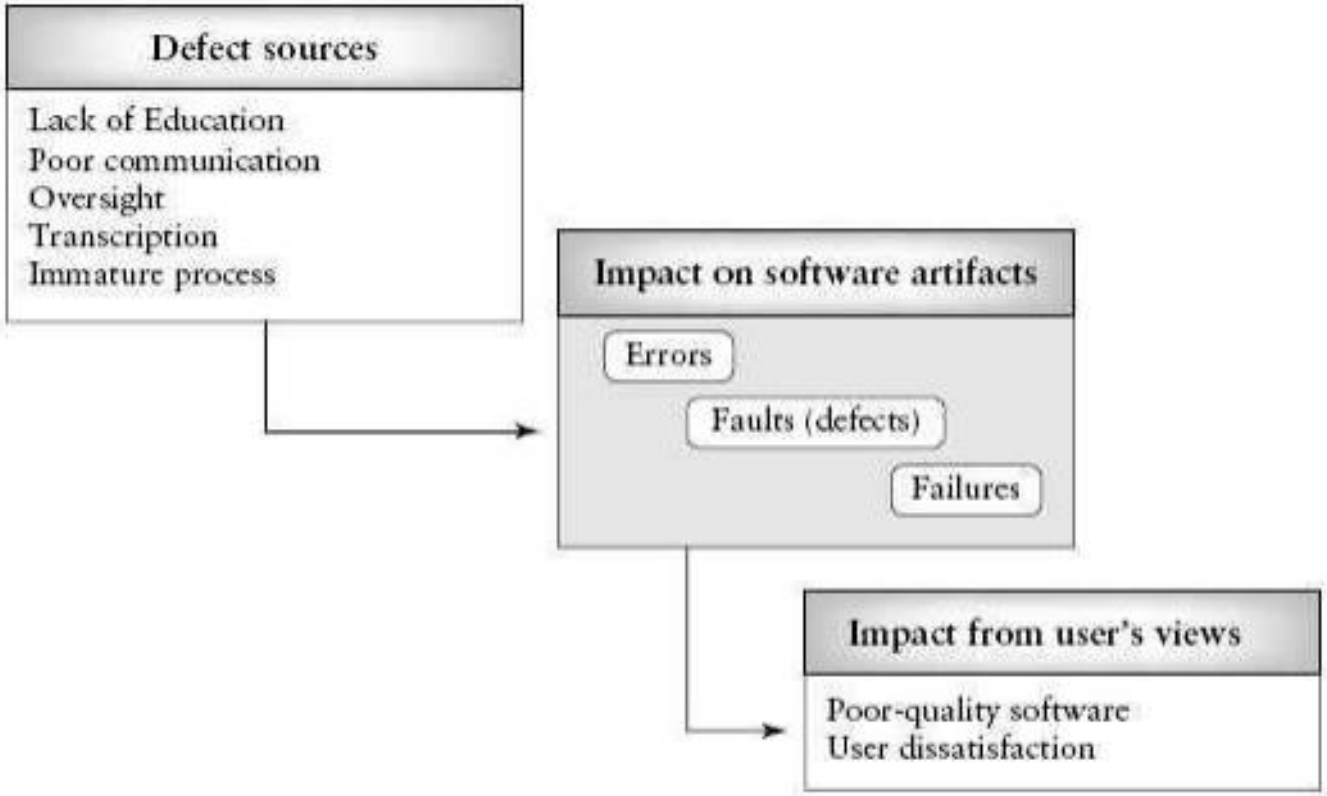
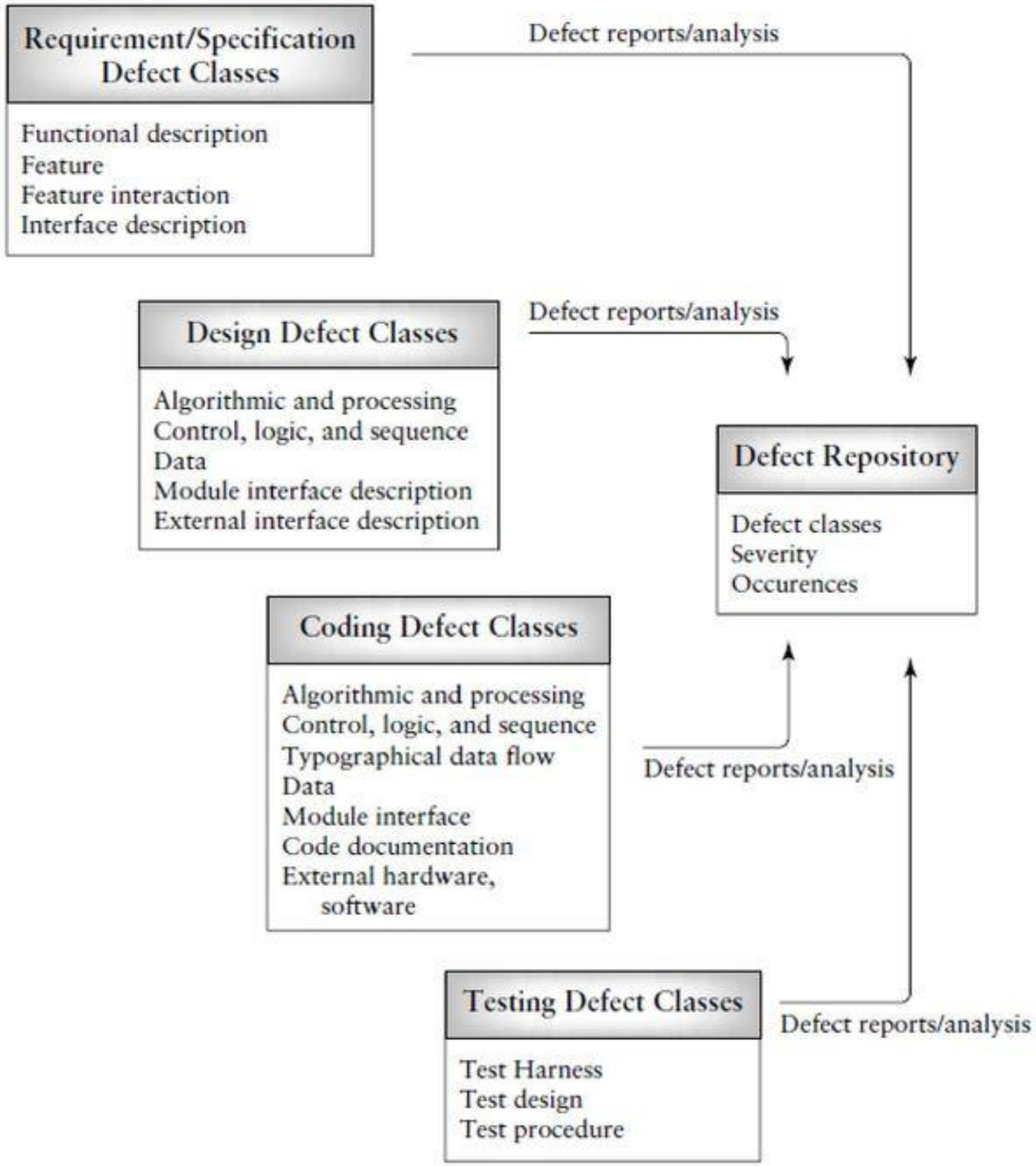


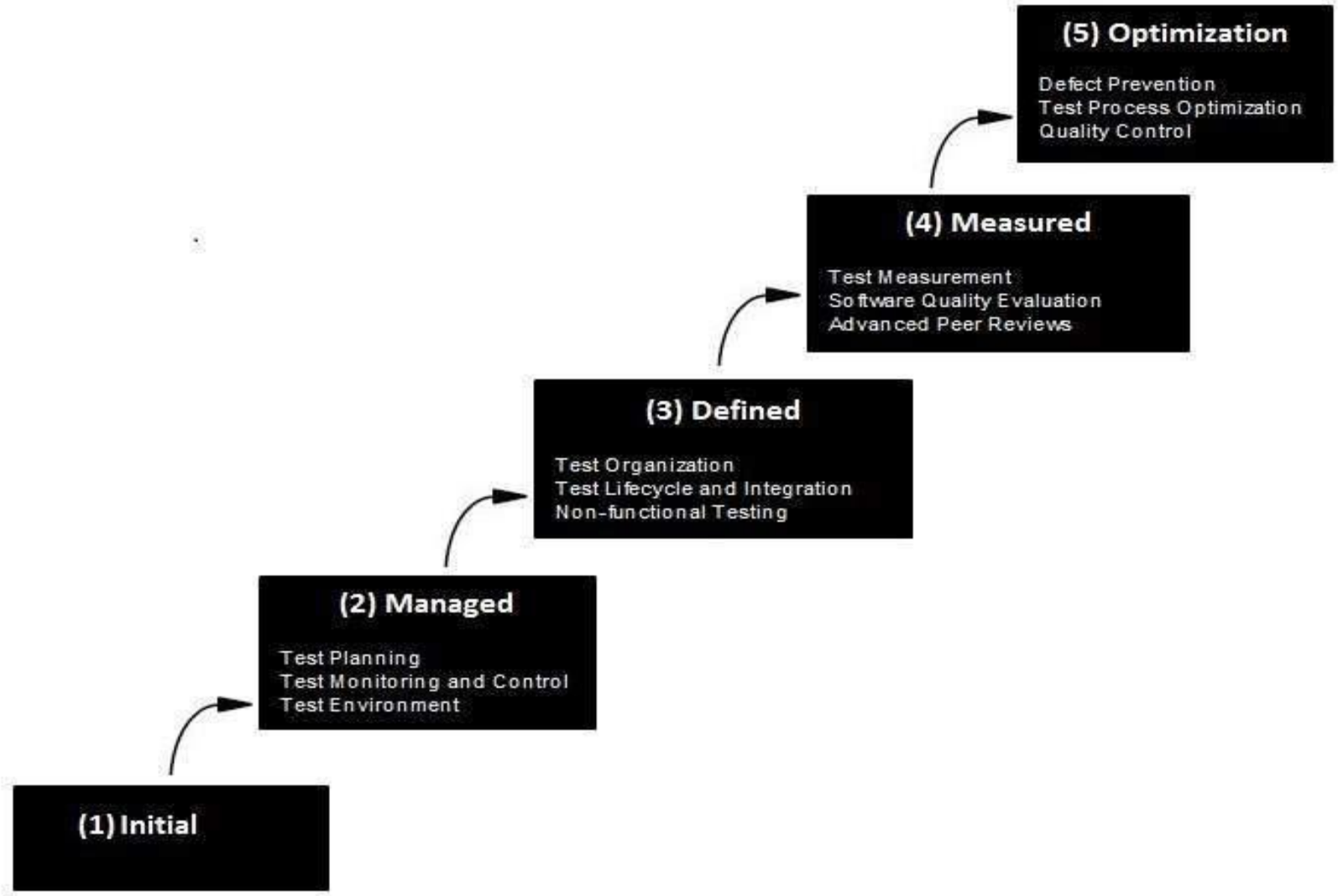
FIG. 3.1

Origins of defects.

Defect classes and the defect repository



Capability Maturity Model (CMM)



Five-Levels of TMM