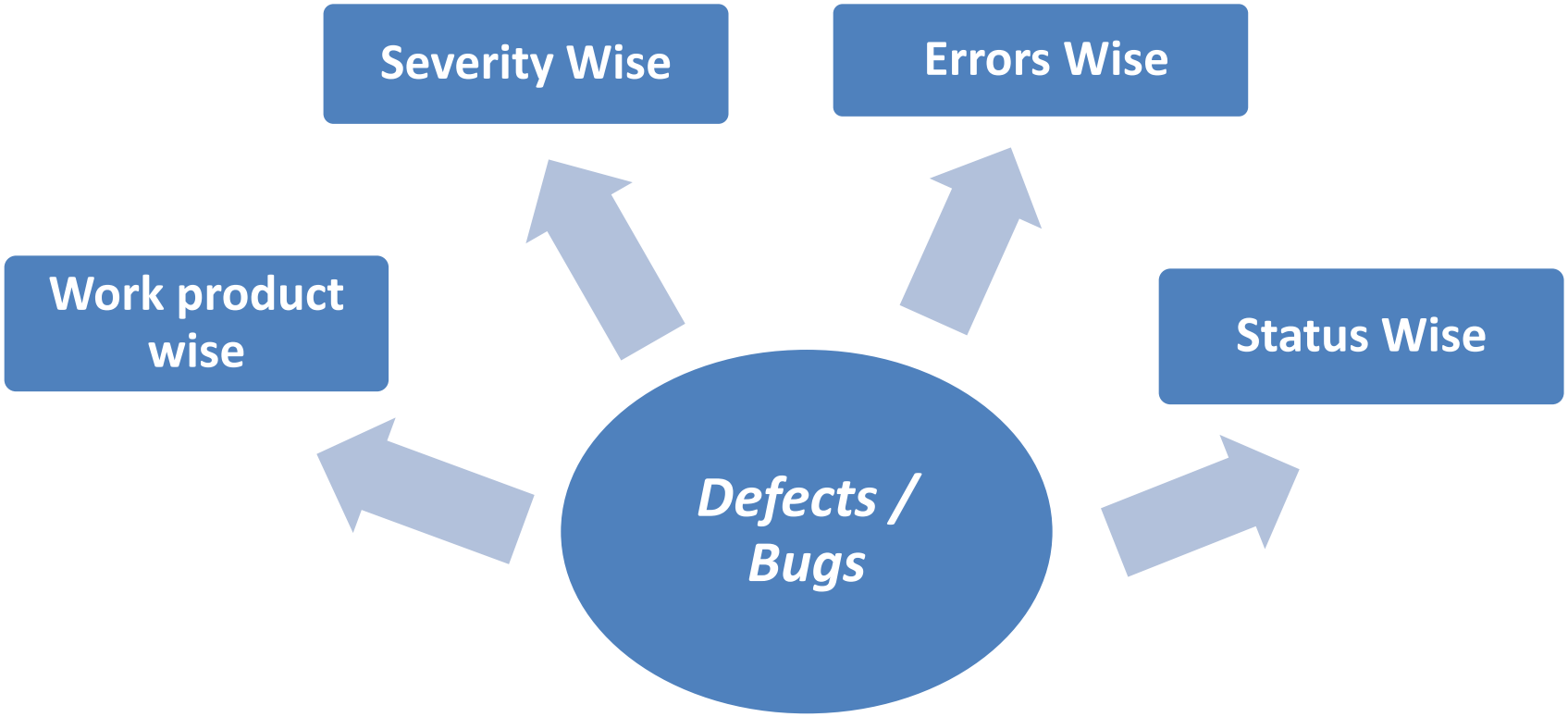
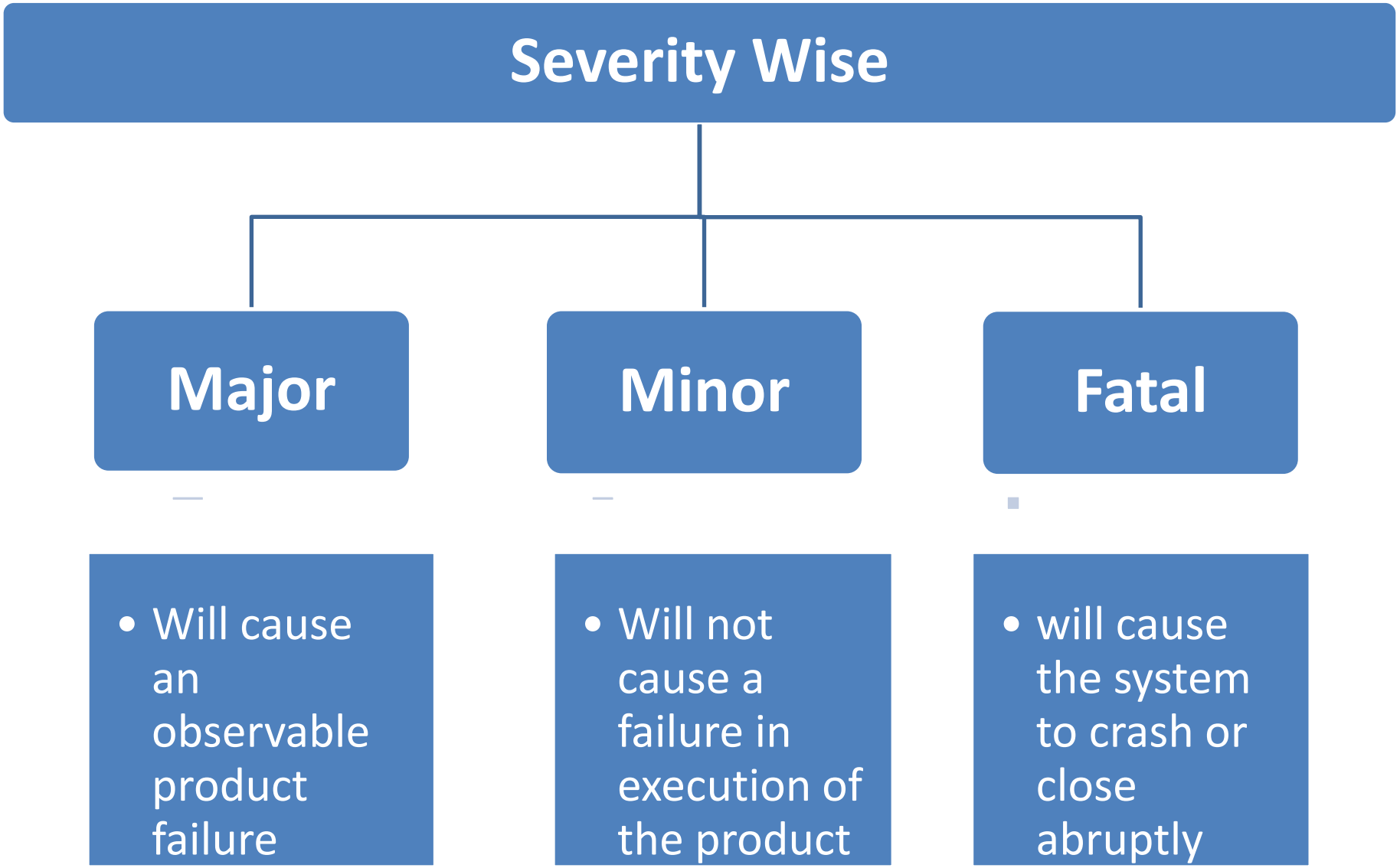
CHAPTER 5: DEFECTS MANAGEMENT

WHAT IS DEFECT ?

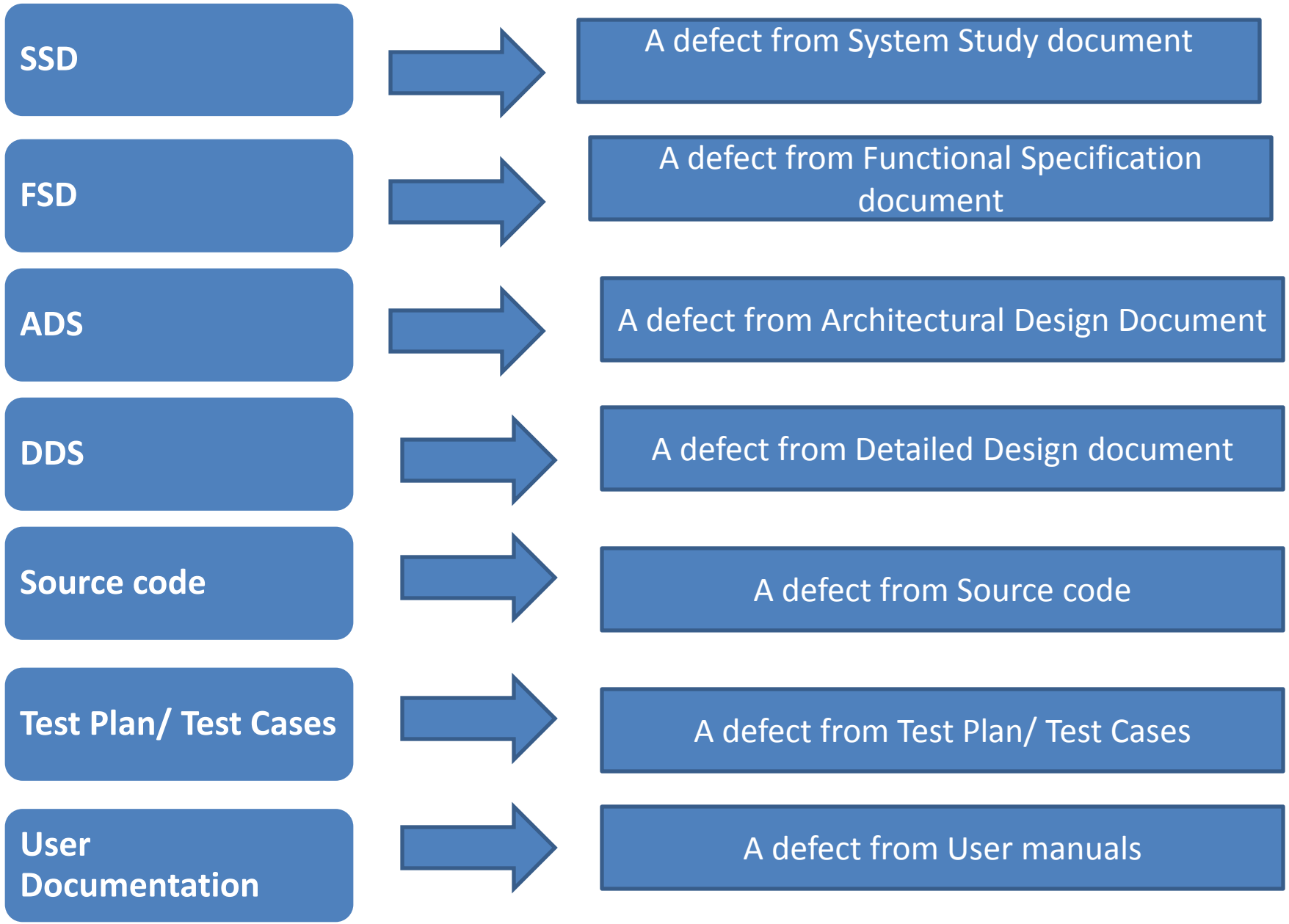
- Defect is basically the difference between the expected result and the actual result.
- A defect is a specific concern about the quality of an Application under Test (AUT).

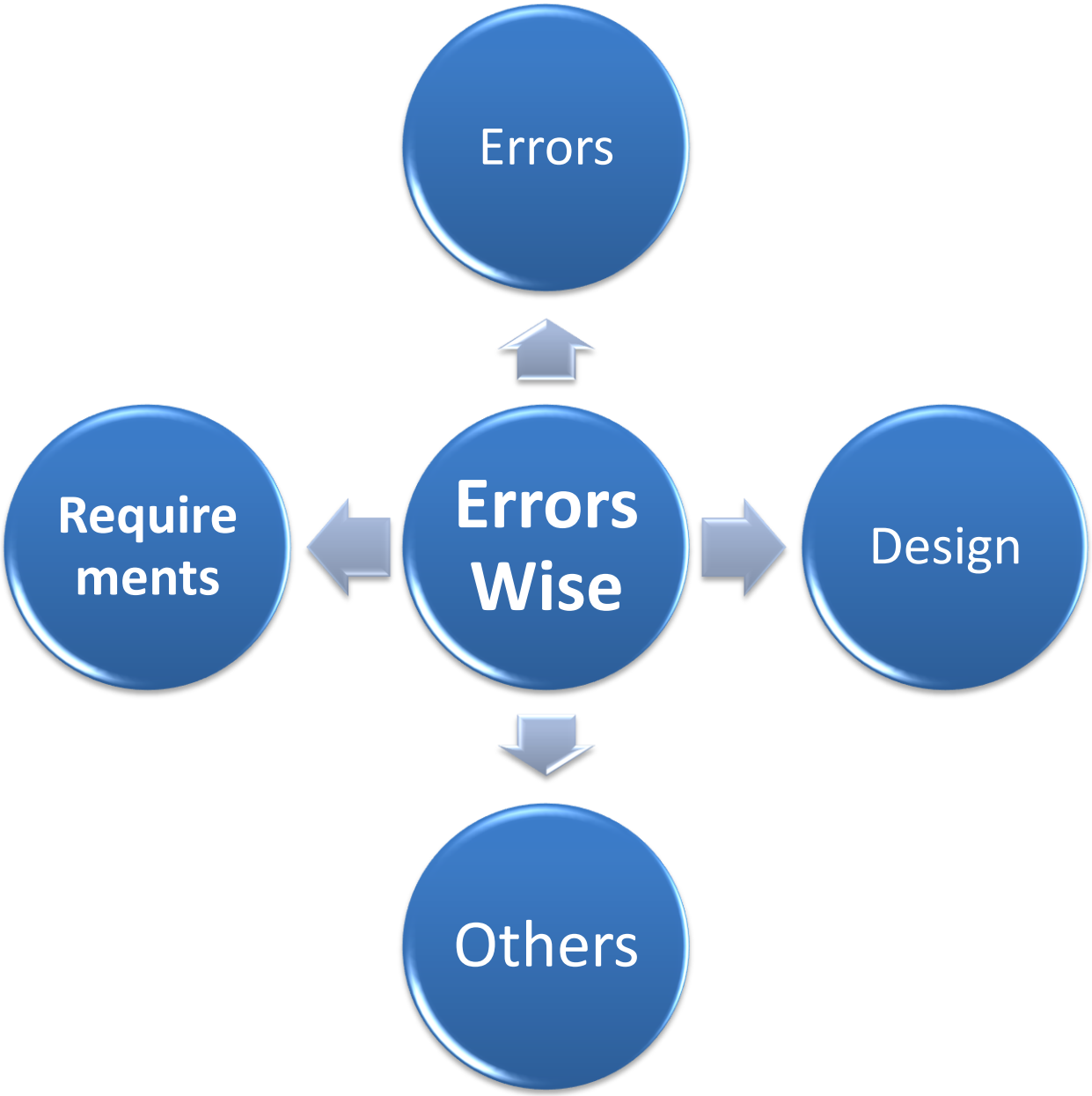
CLASSIFICATION OF DEFECTS / BUGS




Severity Wise
Major
Minor
Fatal
• Will cause an observable product failure
• Will not cause a failure in execution of the product
• will cause the system to crash or close abruptly

Work product wise




Errors
Require ments
Errors Wise
Design
Others

Errors

Navigation Error: Navigation not coded correctly in source code

Message Error: Inadequate/ incorrect/ misleading or missing error messages in source code

Computational Error: Improper computation of the formulae

Data error: Incorrect data population

Database Error: Error in the database schema

Interface Error: Internal or external to application interfacing error Incorrect handling of passing parameters, Incorrect alignment

Logic Error: Missing or Inadequate or irrelevant

Performance Error: An error related to performance/optimality of the code

Sequencing / Timing Error: Error due to incorrect/missing consideration to timeouts

System Error: Hardware and Operating System related error, Memory leak

Test Plan / Cases Error: Inadequate/ incorrect/ ambiguous or duplicate or missing - Test Plan/

Typographical Error: Spelling / Grammar mistake in documents/source code

Variable Declaration Error: Improper declaration / usage of variables

Requirements

Missing Requirements: Implicit/Explicit requirements are missed

Inadequate Requirements: Requirement needs additional inputs for to be complete

Incorrect Requirements: Wrong or inaccurate requirements

Ambiguous Requirements: Requirement is not clear to the reviewer. Also includes ambiguous use of words

Design

Missing Design: Design features/approach missed/not documented in the design document

Inadequate or sub optimal Design: Design features/approach needs additional inputs for it to be completeDesign

In correct Design: Wrong or inaccurate Design

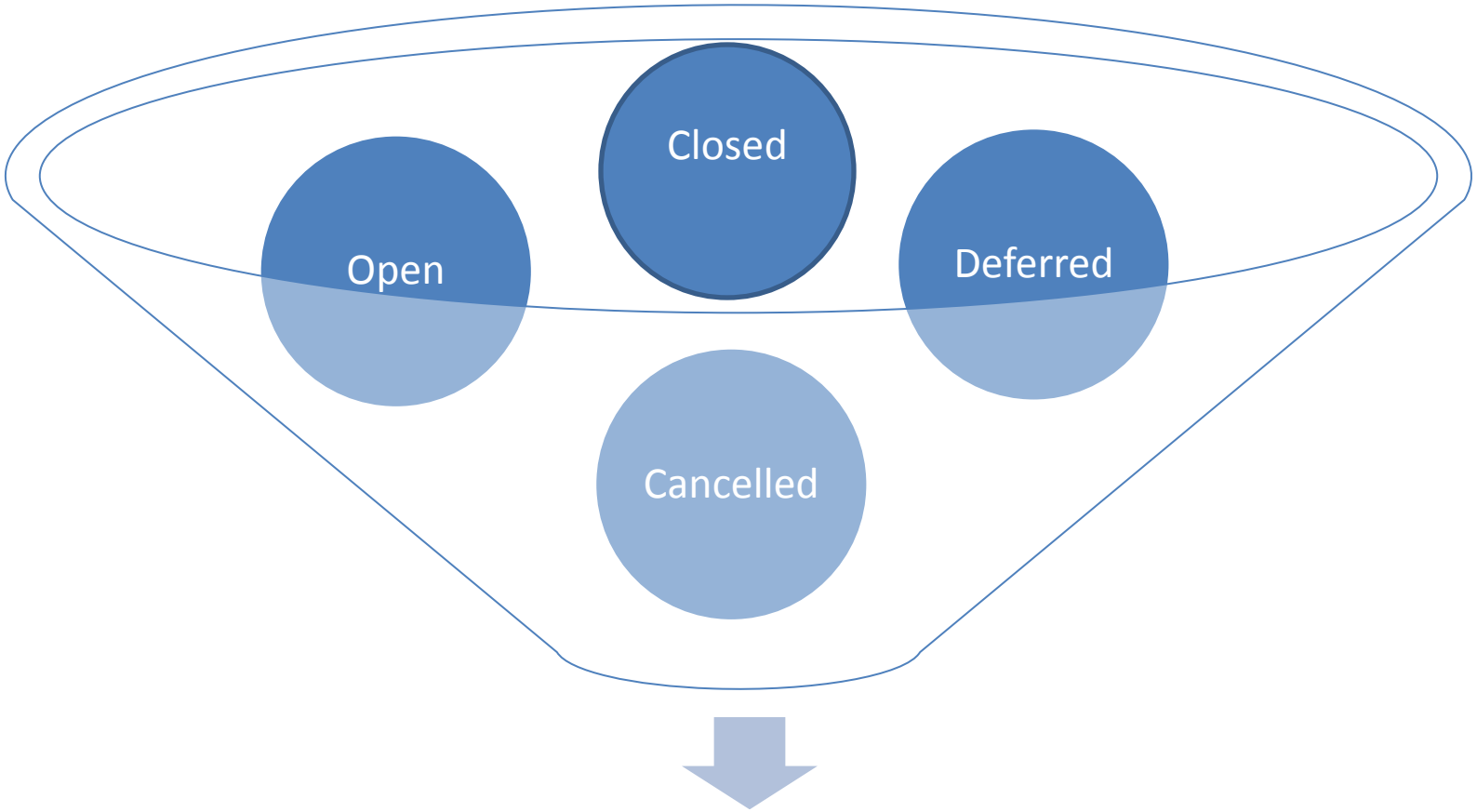
Ambiguous Design: Design feature/approach is not clear to the reviewer

Others

Comments: Inadequate/ incorrect/ misleading or missing comments

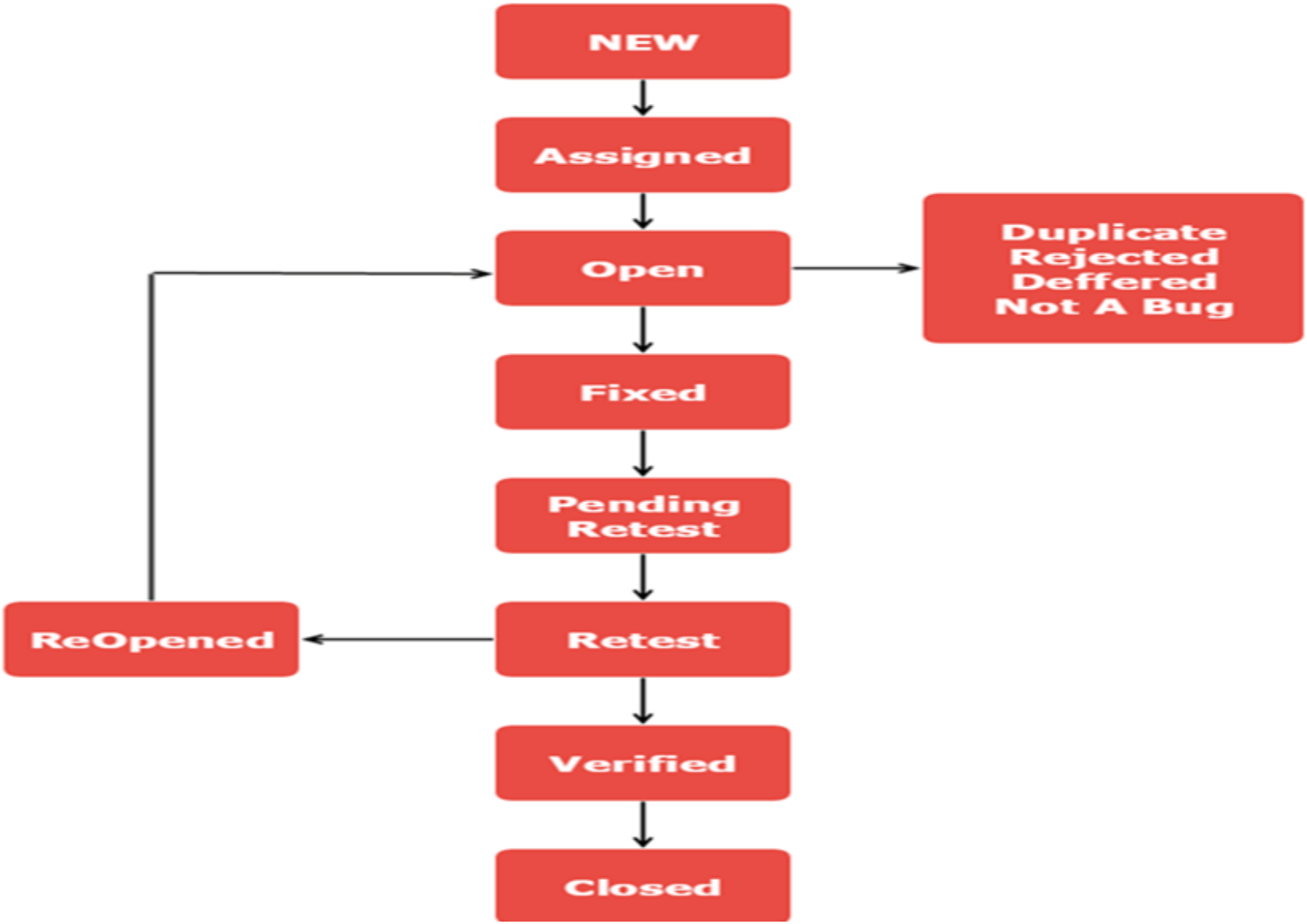
Boundary Conditions Neglected: Boundary conditions not addressed/incorrect

Standards: Standards not followed like improper exception handling, use of E & D Formats



Status Wise

Defect/Bug Life Cycle

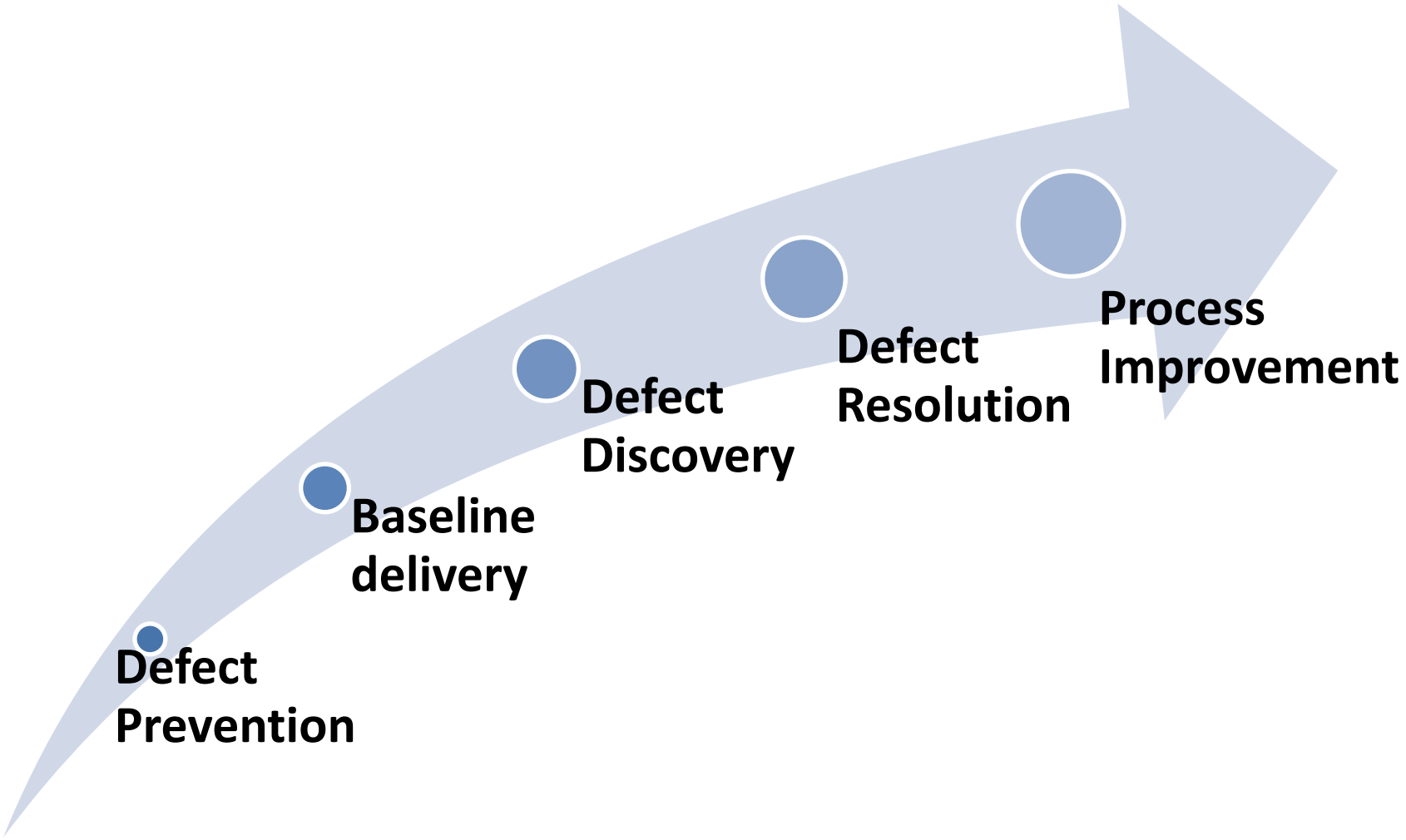


- **New:** When a new defect is logged
- **Assigned:** Assigns the bug to developer team
- **Open**: The developer starts analyzing and works on the defect fix
- **Fixed**: When developer makes necessary code change and verifies the change, he or she can make bug status as "Fixed.“
- **Pending retest**: Code for retesting the code. The status assigned is "pending request."
- **Retest**: To check whether the defect is fixed by the developer or not and change the status to "Re-test."
- **Verified**: . If there is no bug detected in the software, then the bug is fixed and the status assigned is "verified."
- **Reopen**: Once again the bug goes through the life cycle
- **Closed**: If the bug is no longer exits then tester assign the status "Closed.“
- **Duplicate**: If the defect is repeated twice
- **Rejected**: If the developer feels the defect is not a genuine defect
- **Deferred**: If the present bug is not of a prime priority
- **Not a bug:** If it does not affect the functionality of the application

Defect management Process

- Defect is basically the difference between the expected result and the actual result.
- A defect is a specific concern about the quality of an Application under Test (AUT).

There are following parts to manage the process:



Defect Prevention

- It is a process of improving quality and productivity by preventing the injection of defects

Objective

- Identify and analyze the causes of defect
- Reduction in number of defect category
- Reduce most frequent type of defects
- Set goals for improving critical processes
- Reduction of defect escapes between the test phase and the release

Baseline delivery

- A predefined milestone is finished then product is baselined

Terms

- A product should be considered baselined when developers pass it to the testes for testing

Defect Discovery

- It is brought to the attention of the developers and acknowledged to be valid one

Terms

- As soon as team finds the defects, they should report them
- Should be acknowledged by developers
- Should be valid one

Defect Resolutions

- Established for use in the process where there is a dispute regarding a defect

Terms

- A senior manager of the software department will be selected to resolve the dispute
- The problem should be discussed with the product owner to determine whether or not the problem is a defect.

- **Process Improvement:**
- Everyone has different view on process improvement. Developer thinks that defect is not a big deal, but there was a defect it was a big deal.

Some goals of Defect Management Process

- Senior management must understand, support, and be a part of the Defect Management Program.
- Should be integrated into the overall software development process
- Use automated script for the project
- Specific development approaches (e.g., testing, inspections, reviews, exit criteria etc.)
- Improve the process should be driven by the measurement process