SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NAAC – UGC with 'A' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE NAME : 19CS511 SOFTWARE TESTING

III YEAR / V SEMESTER

Unit 1- INTRODUCTION

Topic 7: Origins of Defects and Cost of defects

Origins of Defects and Cost of defects -**Problem**

- An issue can be a bug or a defect. The term issue is often used with clients to indicate that the problem not necessarily is faulty code.
- A defect can also be a bug, but for me a defect is a piece of code that works- but doesn't do what it is expected to do.

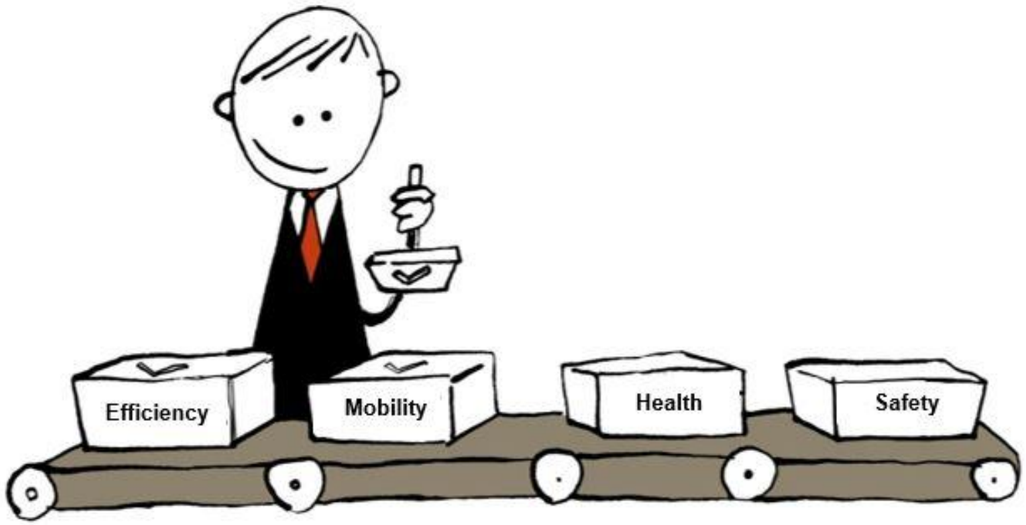
*

What is a Defect?

- A software bug arises when the expected result don't match with the actual results. It can also be error, flaw, failure, or fault in a computer program.
- Most bugs arise from mistakes and errors made by developers, architects.

Defect-Cont..

- Defect can be classified in many ways.
- Defects are assigned to four major classes reflecting their point of origin in the software life cycle.
- These classes are:-
  - Requirements / Specification Defect Classes
  - Design Defect Classes
  - Code Defect Classes
  - Testing Defect Classes



*

Defect -Cont..

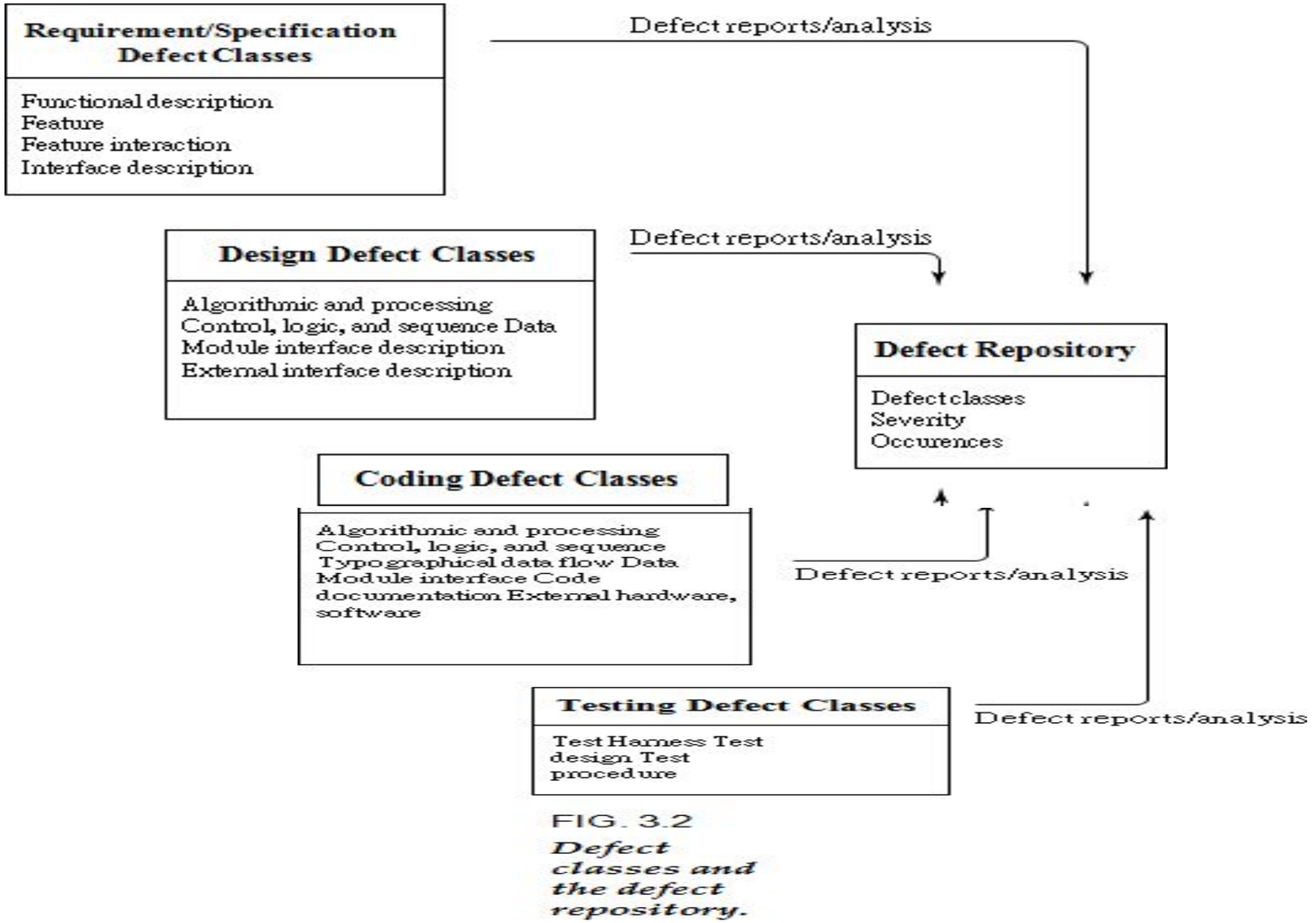


FIG. 3.2
Defect classes and the defect repository.

Requirements and Specification Defects

- **Functional Description Defects:** The overall description of what the product does, and how it should behave.
- **Feature Defects:** Features may be described as distinguishing characteristics of a software component or system.
  - Features refers to functional aspects of software that map to functional requirement described by the user and the client, it also maps quality such as performance and reliability.
  - Feature defects are mainly due to features description that are missing, incorrect, incomplete.

Requirements and Specification Defects-Cont..

3.Feature Interaction Defects:

These are due to an incorrect description of how the features should interact. For example, suppose one feature of a software system supports adding a new customer to a customer database.

4.Interface Description Defects

These are defects that occur in the description of how the target software is to interface with external software, hardware, and users. For detecting many functional description defects, black box testing techniques, which are based on functional specifications of the software, offer the best approach.

Design Defects

1.Algorithmic and Processing Defects

These occur when the processing steps in the algorithm as described by the pseudo code are incorrect.

2.Control, Logic, and Sequence Defects

Control defects occur when logic flow in the pseudo code is not correct. For example, branching to soon, branching to late, or use of an incorrect branching condition.

Design Defects -Cont..

3.Data Defects

These are associated with incorrect design of data structures. For exam- ple, a record may be lacking a field, an incorrect type is assigned to a variable or a field in a record, an array may not have the proper number of elements assigned, or storage space may be allocated incorrectly

4.Module Interface Description Defects

These are defects derived from, for example, using incorrect, and/or in-consistent parameter types, an incorrect number of parameters, or an incorrect ordering of parameters

Design Defects -Cont..

6.External Interface Description Defects

These are derived from incorrect design descriptions for interfaces with COTS components, external software systems, databases, and hardware devices (e.g., I/O devices). Other examples are user interface description defects where there are missing or improper commands, improper sequences of commands, lack of proper messages, and/or lack of feedback messages for the user.

*

Common Types of Defects

 Following are the common types of defects that occur during development:

✔Arithmetic Defects

✔Logical Defects

✔Syntax Defects

✔Multithreading Defects

✔Interface Defects

✔Performance Defects

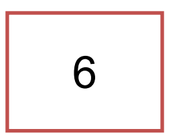
Example Program

```
#include <stdio.h>
int main()
{
int number1,  sum;
printf("Enter two integers: ")
scanf("%d ", &number1, &number2);
// calculating sum
sum = number1 + number3:
printf("sum);
return 0;
)
```

*

Defect Identification

```
#include <stdio.h>
int main()
 {
int number1, number2, sum;
printf("Enter two integers: ");
scanf("%d %d", &number1, &number2);
 // calculating sum
 sum = number1 + number2;
printf("%d + %d = %d", number1, number2, sum); return 0;
}
```




Set Up/Modify Defect Repository
Train Users
Ensure Defects are Entered Correctly
Prioritize and Assign
Verify Timely Refactoring
Track and Report
Identify and Relate Defect Trends
Defect Management

Activity

Advantages and Disadvantages

Advantages	Disadvantages
• Simple model and easy to manage • Applicable for small software	• "Big Design Up Front" • Defect detected at late phases • High amounts of risk and uncertain
• Early testing involvement • Clear relationship between test phases development phases	• Still possess limitation of sequential model • Require high amount of documentation • Duplication of testing effort
• Risk and uncertain are managed • Testing activities and process are managed	• Heavy documentation • Late customer involvement – only at UAT
• Adaptable to changes • Early client involvement - Avoid unrealistic requirements • Avoid spending time on useless activities	• Require high-capable people • Need representative from client • Problem scaling up the architecture

Assessment 1

1. List out the Advantages of defect

a)_____

b)_____

c)_____

d)_____

2. Identify the Disadvantages of defect

a)_____

b)_____

c)_____

d)_____

TEXT BOOKS:

1. Ricardo Baeza-Yates and Berthier Ribeiro-Neto, —Modern Information Retrieval: The Concepts and Technology behind Search, Second Edition, ACM Press Books, 2011.

2. Ricci, F, Rokach, L. Shapira, B.Kantor, —Recommender Systems Handbook‖, First Edition, 2011.

REFERENCES:

1. C. Manning, P. Raghavan, and H. Schütze, —Introduction to Information Retrieval, Cambridge University Press, 2008.

2. Stefan Buettcher, Charles L. A. Clarke and Gordon V. Cormack, —Information Retrieval: Implementing and Evaluating Search Engines, The MIT Press, 2010.

THANK YOU