CHAPTER 4:

HIGH QUALITY DESIGN

4.1 HIGH QUALITY DESIGN: REUSABILITY

- As per the inputs based on the experience of Mr. Pramod Ghorpade, Dev Manager at NetApp, Bengaluru; Mr. Vinod Ghorpade, Senior Practice Manager at Wipro Technologies, Reading, United Kingdom (UK); Ms. Vasudha C, Module Leader at Wipro Technologies, Bengaluru

- Reuse is the only currently available technology that shows promise of order of magnitude improvements in software development quality and productivity.

- One of the principal reasons this work spends so much time addressing high quality issues is to help us to produce code that is of suitable quality for inclusion in a reuse library.
4.2 CONSISTENT AND ACCURATE SIZE MEASUREMENT DATA

- As per the discussion and inputs from Mr Sachin Punadikar, Senior Staff Software Engineer, NAS Team, India Software Lab, IBM India. Ozone2, Pune.

- Once we have selected a precise size measure, we will need automated means to ensure it is accurately and economically counted.

- Manually counting the instructions in even small programs is tedious, time consuming, and inaccurate.

- For large programs, it is practically impossible.

- In the late 1980s, IBM had an inventory of over 30,000,000 LOC and shipped several million new LOC every year. Calculate for 2012?

- It used a single standard automated counter to ensure that every laboratory produced consistent and accurate size measurement data.
4.3 REVIEWS AND HIGH QUALITY DESIGN

- Defect data gathered on both during industry internship and experienced engineers (Information provided by the corporate Professionals like Mr. Sachin, Mr. Vinod and others from Wipro, IBM) shows that

- Typically the number of compile and unit test defects falls between 50 to 200 per 1 KLOC

- This range is for undisputed and unrelieved programs.

- With comprehensive code reviews, fewer defects are found in compile and test;

- however, more are found overall, then the range is more like 75 to 200 defects per KLOC.

- This is one defect for every five to twelve lines of code.

- To manage our defects, we must know how many we make.

- We do this by gathering and analyzing the defects found in compile and test for our earlier programs.

- With these data, we can establish review checklists and make our own process quality assessments
4.4 DEFECTS: CORPORATE SCENARIO

• If there were only one or two defects in a one-KLOC program, we might question the wisdom of studying every line to ensure its correctness.

• Of course, whether we do study every line depends on how important it is that the program be defect free.

• The discussion and inputs from Mr. Kapil Kumar, Associate Vice President, InfoTel Group, Reliance, New Mumbai; Mr. Vinod Ghorpade, Senior Practice Manager at Wipro Technologies, Reading, United Kingdom (UK); Ms. Vasudha C, Module Leader at Wipro Technologies, Bengaluru; Mr. Sujay Ghorpade, SAP PS/cProjects Consultant, Bengaluru; Ms. Aarti Agarwal, Senior Software Engineer, Pitney Bowes Business Insight, Customer Communication Management, Logix Cyber Park, Noida (U.P) show, however,

• that even experienced programmers typically make about 100 defects per KLOC.*

• * 1st time compilation and Testing
4.5 DEFECTS: EXPERIMENTATION

- Four MBATech IT Interns at various companies produced a total of average of 1200 LOC for their respective modules.

- Their average total defects for these programs were 117, or 97.5 defects per KLOC.

- Of these, 68 were found in compile and 49 in test.

- On these four modules; the engineers' composite defect rate fluctuated between 73 and 123 defects per KLOC.

- In general, however, about half their defects are found in compile; they must find the rest in test.

- This means that unless we are unusually talented, we will have to find around 50 defects per KLOC in test.

- The issue is not how much time we will likely waste looking for a rare defect, but what is the most effective way to find and fix lots of defects.