

What is Defect Density? Formula to calculate with Example

What is Defect Density?

Defect Density is the number of defects confirmed in software/module **during a specific period of operation** or development divided by the size of the software/module. It enables one to decide if a piece of software is ready to be released.

Defect density is counted per thousand lines of code also known as KLOC.

How to calculate Defect Density

A formula to measure Defect Density:

Defect Density = Defect count/size of the release

Size of release can be measured in terms of a line of code (LoC).

Defect Density Example

Suppose, you have 3 modules integrated into your software product. Each module has the following number of bugs discovered-

- Module 1 = 10 bugs
- Module 2 = 20 bugs
- Module 3 = 10 bugs

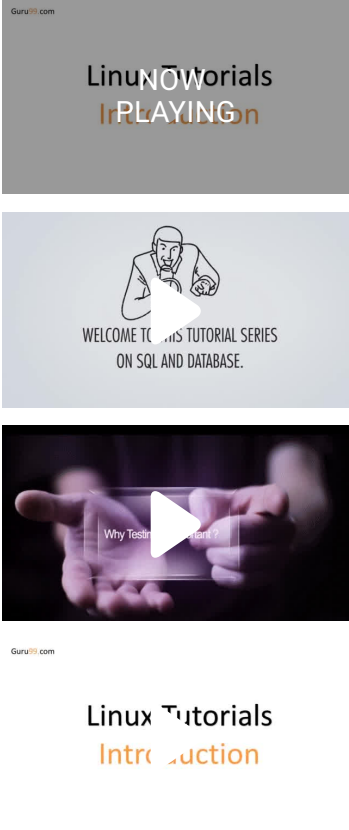
Total bugs = 10+20+10 =40

The total line of code for each module is

- Module 1 = 1000 LOC
- Module 2 = 1500 LOC
- Module 3 = 500 LOC

Total Line of Code = 1000+1500+500 = 3000

FEATURED VIDEOS



Defect Density is calculated as:

Defect Density = 40/3000 = 0.013333 defects/loc = 13.333 defects/KLoC



A standard for defect density

However, there is no fixed standard for bug density, studies suggest that one Defect per thousand lines of code is generally considered as a sign of good project quality.

Factors that affect the defect density metrics

- Code complexity
- The type of defects taken into account for the calculation
- Time duration which is considered for Defect density calculation
- Developer or Tester skills

Advantages of defect density

- It helps to measure the testing effectiveness
- It helps to differentiate defects in components/software modules
- It is useful in identifying the areas for correction or improvement
- It is useful in pointing towards high-risk components
- It helps in identifying the training needs to various resources
- It can be helpful in estimating the testing and rework due to bugs
- It can estimate the remaining defects in the software
- Before the release, we can determine whether our testing is sufficient
- We can ensure a database with a standard defect density

Prev

Report a Bug

Next

YOU MIGHT LIKE:

SOFTWARE TESTING

Test Plan Template: Sample Document with Web Application Example
What is test plan template? TEST PLAN TEMPLATE is a detailed document that describes the test...
[Read more »](#)

SOFTWARE TESTING

Salesforce Testing Tutorial: What, Process, Tools, Best Practices
What is Salesforce? Salesforce is the world's first cloud-based CRM system. It was founded by Marc...
[Read more »](#)

AGILE TESTING

Agile Vs Waterfall: Know the Difference Between Methodologies
What is Waterfall methodology? Waterfall Model methodology which is also known as Liner Sequential Life...
[Read more »](#)

COURSE

Live Penetration Testing Project: Online Practice in Real Time
Project Summary Security is the most dreaded word in IT industry. Its not uncommon to hear news of a...
[Read more »](#)

SDLC

Incremental Model in SDLC: Use, Advantage & Disadvantage
What is Incremental Model? Incremental Model is a process of software development where...
[Read more »](#)

COURSE

Free Software Testing Live Project: Insurance Domain
Project Summary The BFSI (Banking, Financial services and Insurance) sector is the biggest...
[Read more »](#)

About
[About Us](#)
[Advertise with Us](#)
[Write For Us](#)
[Contact Us](#)

Career Suggestion
[SAP Career Suggestion Tool](#)
[Software Testing as a Career](#)

Interesting
[eBook](#)
[Blog](#)
[Quiz](#)
[SAP eBook](#)

Execute online
[Execute Java Online](#)
[Execute Javascript](#)
[Execute HTML](#)
[Execute Python](#)

Top Tutorials

Selenium

Testing

Hacking

SAP

Java

Python

Jmeter

Informatica

JIRA