1. INTRODUCTION :

1.1 Objective of proposed system :

Defect Tracking for Improving Product Quality and Productivity for Improving Software Reliability is an automated system that can be useful to employees and the managers in any functional organization. Defect Tracking for Improving Product Quality and Productivity gives the facility to define the tasks in the organization and also allows the managers to track the Defects spent by the developer for that particular task. A report generation facility is supported in DTS that allows the managers to analyze which are those skills by employee are utilized and those which are not utilized. project or application. This tool helps employees to document their Defects and analyze.

1.2 Proposed system :

The Proposed system is a browser which is completely related to online system, which provides the centralized database. It stores Defects data and description of the particular Defect data. It can also create reports and documents based on the information in its database.

1.3 Details of project :

**Existing System :**

The existing system consists of entering the details in the Microsoft Excel Sheets for the storing of the data. When a manager needs information of the employee he searches for the specified file in the file system. He opens the file and takes the information. Report Generation done manually by copying the content of the different files into another file. The Manually generated report was then printed.

**Limitations in Existing System :**

* Information retrieval is a very big process.
* Lack of organization of the files may porn to information loss due to accidental deletion of files.
* No security because the files are visible to the users.
* Report generation will be a big task.

**: Advantages over Existing System :**

* The performance is increased due to well designed database.
* Security is increased
* Time saving in report generation
* Easy to update the details

Need of the system :

[Tracking defects](http://searchsoftwarequality.techtarget.com/answer/When-tracking-defects-make-efficiency-the-end-goal) found in testing as a performance metric tends to be demotivating, placing software testers and developers in an adversarial position when they should be in a cooperative position. Instead, if the purpose of [tracking defects is to look for patterns](http://searchsoftwarequality.techtarget.com/tip/Tracking-bugs-effectively-in-continuous-development) in software development, including requirements definition, design and coding, where problems are found on a regular basis, this may provide information around aspects that need improvement in the organization.

1. Technology & Methodology used :
   1. H/W and S/W Requirement:

**Software Requirements**

1. Operating System : Window XP/2003orLinux/Solaris
2. User Interface : HTML, CSS
3. Client-side Scripting : JavaScript
4. Programming Language : Java
5. Web Applications : JDBC, JNDI, Servlets, JSP
6. IDE/Workbench : MyEclipse8.6
7. Database : Oracle10G
8. Server Deployment : Tomcat6.0

**Hardware Requirements**

1. Processor : Pentium IV
2. Hard Disk : 40GB
3. RAM : 256MB

2.3 **SDLC METHODOLOGIES :**

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

* The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
* A preliminary design is created for the new system.
* A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
* A second prototype is evolved by a fourfold procedure:

1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
2. Defining the requirements of the second prototype.
3. Planning an designing the second prototype.
4. Constructing and testing the second prototype.

* At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involved development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer’s judgment, result in a less-than-satisfactory final product.
* The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
* The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
* The final system is constructed, based on the refined prototype.
* The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

**The following diagram shows how a spiral model acts like:**



**Fig 1.0-Spiral Model**

**ADVANTAGES**

* Estimates(i.e. budget, schedule etc .) become more relistic as work progresses, because important issues discoved earlier.
* It is more able to cope with the changes that are software development generally entails.
* Software engineers can get their hands in and start woring on the core of a project earlier.

**3. Design & Implementation :**

**3.1 SYSTEM DESIGN**

System design of defect tracking for improving product quality and productivity (D.T.S) contains the various types of tables of records that we can understand from the ER-daigarm and Data flow diagram(D.F.D) with the collection of defferent levels DFD 1,2 .

3.2 Flow chart: