



**Varun C.M.**

Assistant Professor

St. Xavier's Catholic College of Engineering

# Software

An accumulation of **programming language instructions and statements or development tool instructions** that together form a program or software package. This program or software package is usually referred to as the “**code**”.

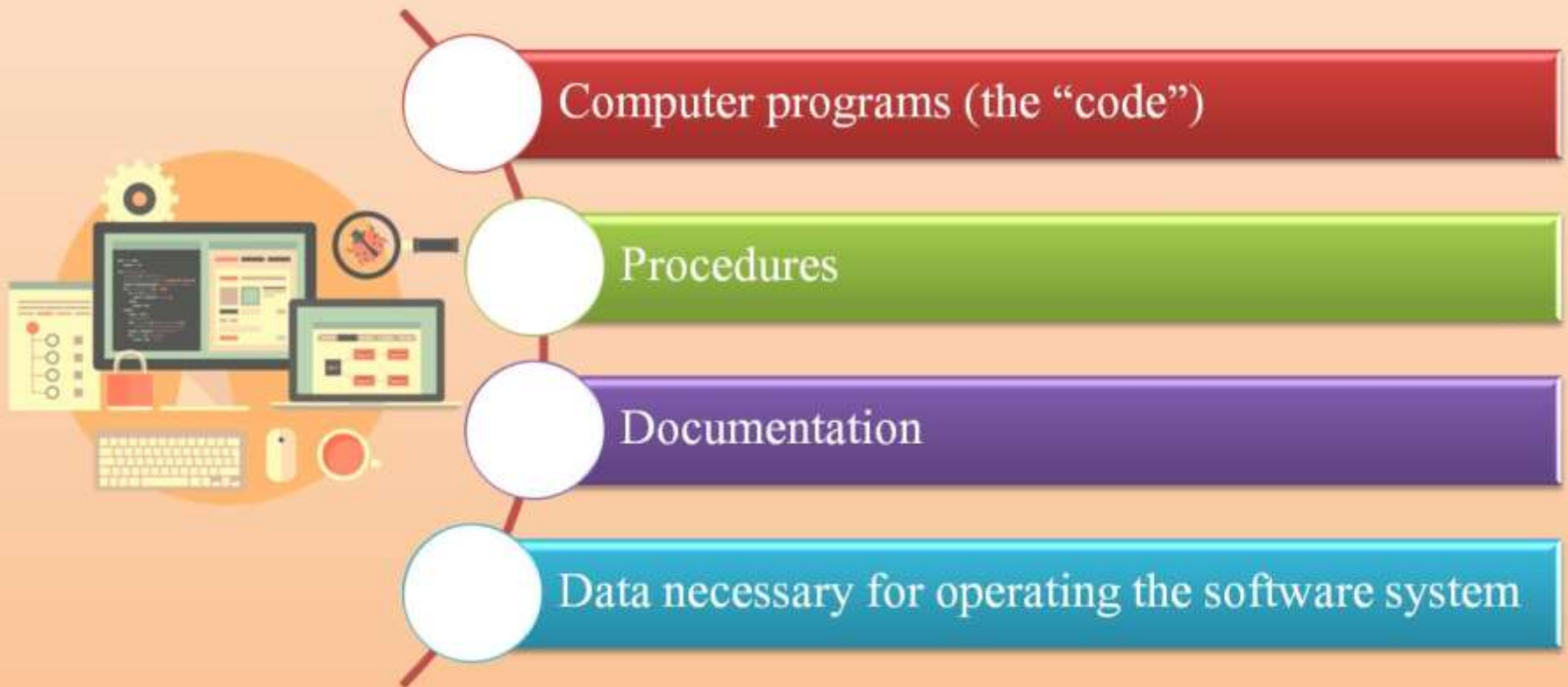
## Software - IEEE definition

Software is:

Computer programs, procedures, and possibly associated documentation and data related to the operation of a computer system.

# Software

All four components are needed in order to assure the quality of the software development process and maintenance services



# Software



Computer programs (the “code”)

**Computer programs (the “code”) are needed because they activate the computer to perform the required applications**

# Software



## Procedures

**Procedures are required, to define the order and schedule in which the programs are performed**



# Software



## Documentation

**Various types of documentation are needed for developers, users and maintenance personnel**

# Software




## Documentation

### **The development documentation**

Allows efficient cooperation and coordination among development team members

### **The user's documentation**



Provides a description of the available applications and the appropriate method for their use

### **The maintenance documentation**

Provides the maintenance team with all the required information

# Software



Data necessary for operating the software system

**Data including parameters, codes and name lists are necessary for operating the software to satisfy the need of specific user.**

**The standard test data, to find out the expected software malfunctioning**



# Software Errors, Faults and Failures



“We’ve used the Simplex HR software in our Human Resources Department for about three years and we have never had a software failure.”



“I started to use Simplex HR two months ago; we had so many failures that we are considering replacing the software package.”



“We have been using the same software package for almost four years. We were very satisfied throughout the period until the last few months, when we suddenly faced several severe failures. The Support Center of the software house from which we bought the package claims that they have never encountered failures of the type we experienced even though they serve about 700 customers who utilize Simplex HR.”

# Software Errors, Faults and Failures



Is it possible for such a variation in users' experience with failure to appear with the same software package?

Can a software package that successfully served an organization for a long period “suddenly” change its nature (quality) and become “bugged”?

The answer to these questions is yes, and it is rooted in the characteristics of software.

# Software Errors, Faults and Failures



**The origin of software failures lies in a software error made by a programmer.**

An error can be a grammatical error in one or more of the code lines, or a logical error in carrying out one or more of the client's requirements.

**Not all software errors become software faults.**

In some cases, the software error can cause improper functioning of the software in general or in a specific application.

In many other cases, erroneous code lines will not affect the functionality of the software as a whole

# Software Errors, Faults and Failures

In a part of these cases, the fault will be corrected or “neutralized” by subsequent code lines.

In many situations, a software fault is never activated due to the user’s lack of interest in the specific application or to the fact that the combination of conditions necessary to activate the fault never occurs.



# Software Errors, Faults and Failures

Example 1:

The “Pharm-Plus” software, a software package developed for the operations required of a pharmacy chain, included several software faults:

- 1.** The chain introduced a software requirement to avoid the current sale of goods to customers whose total debts will exceed \$200 upon completion of the current sale. Unfortunately, the programmer erroneously put the limit as \$500, a clear software fault. However, a software failure never occurred as the pharmacies do not offer credit to their customers, that is, sales are cash sales or credit card sales.



# Software Errors, Faults and Failures

**2.** Another requirement introduced was the identification of “super customers”. The requirement of the software was to identify those customers who made a purchase at least once a month, the average value of that purchase made in the last 12 months being more than 5 times the value of the average customer’s purchase at the pharmacy. It was required that once “super customers” reached the cashier, they would be automatically identified by the cash register. (The customers could then be treated accordingly, by receiving a special discount or gift.)

The software fault (caused by the system analyst) was that “super customers” could be identified only by the value of their current purchase.

It causes the identification of wrong customer as “super customers” and the software fault turned in to software failure.

# Software Errors, Faults and Failures

Example 2:

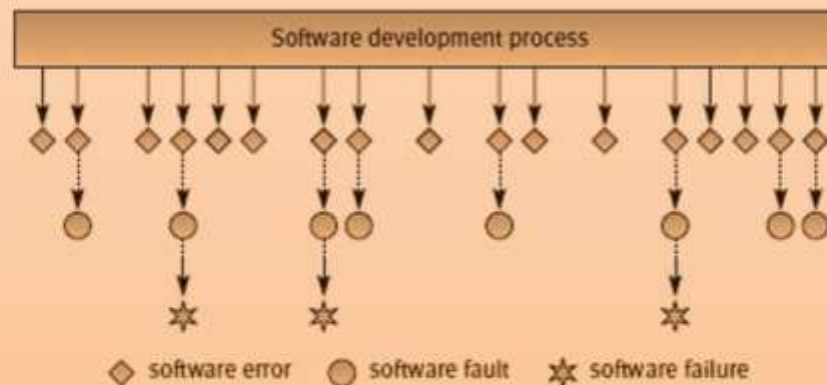
The “Meteoro-X” meteorological equipment firmware .

- The software requirements for “Meteoro-X” meteorological equipment firmware (software embedded in the product) were meant to block the equipment’s operation when its internal temperature rises above 60°C.
- A programmer error resulted in a software fault when the temperature limit was coded as 160°.
- This fault could cause damage when the equipment was subjected to temperatures higher than 60°.
- Since the equipment was used only in some coastal areas where temperatures never exceeded 60°, the software fault never turned into a software failure.

# Software Errors, Faults and Failures

These examples should adequately make the point that **only a portion of the software faults**, and in some cases only a small portion of them, will turn into software failures in either the early or later stages of the software's application.

Other software faults will remain **hidden, invisible** to the software users, yet capable of being activated when the situation changes.





# Summary



**Software Definitions?**



**Components of software?**



**Software Failure**