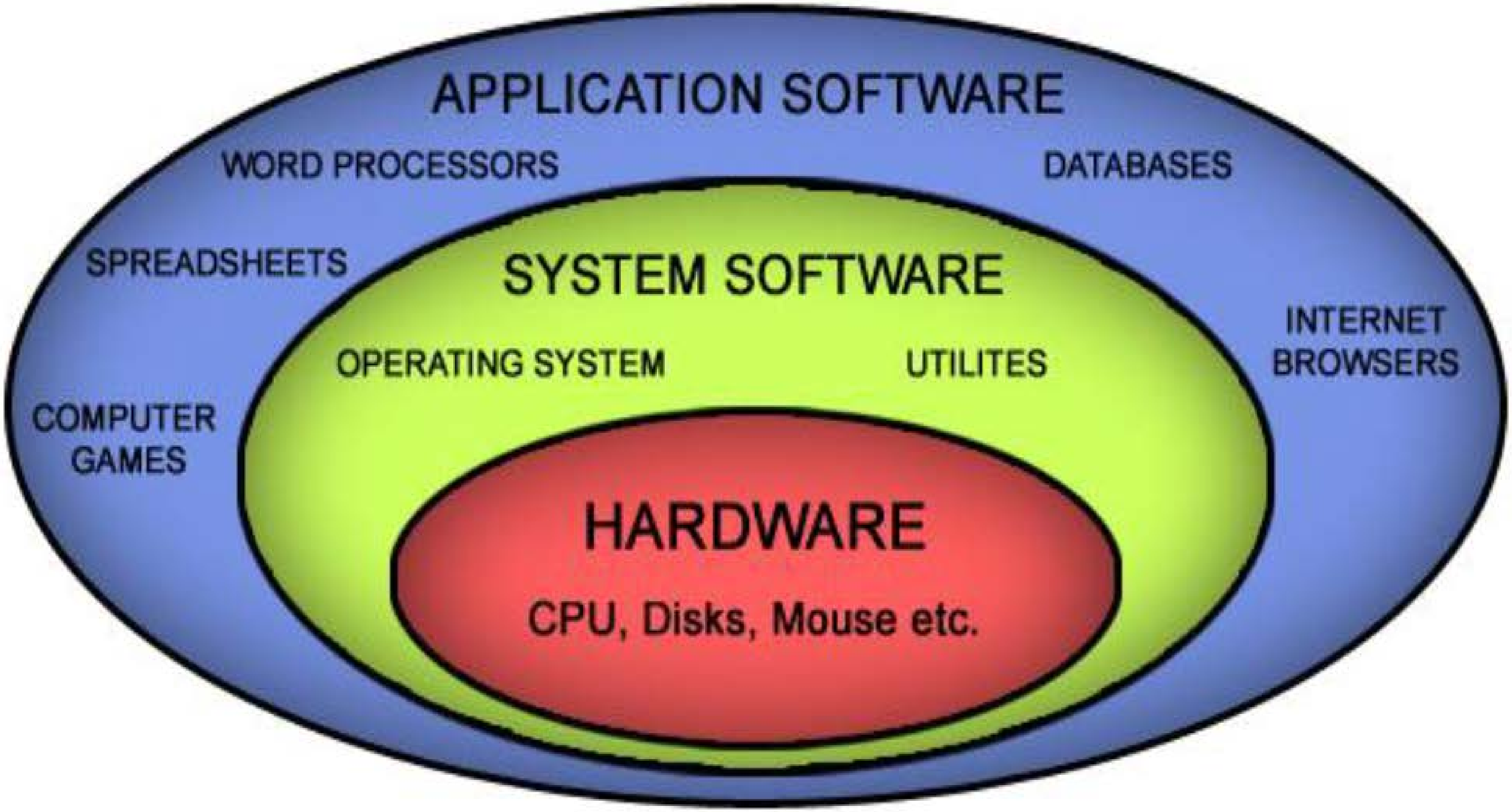
INTRODUCTION TO SOFTWARE

JAGJIT BHATIA

DEPT OF COMPUTER SCIENCE & IT

OCT,2017

- We are all aware of the term software. Software is actually that part of computer, which is not tangible, i.e. which we cannot touch but without it no part of computer can work. It was only when software was invented that the size of computer decreased and the computer became a multi-tasker. So, software is as important as hardware. Let us define software. It is known as a set of programs, which comprise of all the programs, which are executable on the computer, libraries, applications and even the scripts. They are actually a set of clear-cut instructions upon which the computer works.



APPLICATION SOFTWARE

WORD PROCESSORS

DATABASES

SPREADSHEETS

INTERNET BROWSERS

COMPUTER GAMES

SYSTEM SOFTWARE

OPERATING SYSTEM

UTILITES

HARDWARE

CPU, Disks, Mouse etc.

Various Types of Software

- **System Software**: It is the main software, which helps in running the computer system and the entire as well as the hardware. The various examples include servers, window systems, drivers, diagnostic tools etc.
- **Application software**: This kind of software allows the users to achieve more than one computer related specific tasks. If you want to understand it, think of terms like window applications and other software present in current smartphones, tablets etc. (APPS). Other application software includes business software, medical software, designing software, gaming software etc. Web applications give output on web pages e.g. PHP, ASP.NET ETC.
- **Programming Software:** These types of software aid the programmers in writing programs in an easy manner. These include interpreters, compilers, linkers, debuggers etc. Software programmers to create, maintain or debug the computer systems use this type of software. Further, there are embedded software, which are put as firmware in the tiny electronic chips to be used by single users in automobiles.
- **Firmware** can actually be defined as sequence of instructions that is substituted for hardware. Language of Software: Software is usually written in various programming languages. Actually, a programming language is the one, which is used by programmers to develop various applications, scripts and other group of commands. Examples include COBOL, FORTRAN, BASIC, SQL, FOXPRO etc. The table given below will give a birds eye view of the various types of applications and the languages used to develop these software

Table: Different types of languages used for specific types of applications and programs

Types of application/Software	Characteristics	Languages used
Artificial Intelligence Development	Creating character interactions in computer games, chatbots etc.	C,C++, Prolog, AIML
Database Development	Creating and maintaining characters	DBASE, FoxPro
Game Development	Development of computer games and other software	Java
Computer drivers or other hardware interface development	To embed systems and generate device drivers	Assembly, C
Internet and Web Page Development	Creating own web pages, developing internet applications, Internet related tasks	HDML, HTML, Java, JavaScript, Perl
Script Development	To create and develop scripts to increase performance	Auto hotkey, Perl

Platform software

- A platform in the fundamental unit of the computer hardware, Which integrates software and on which the software applications are run. It includes the software, which help the user to form a link between computer and its allied parts. Such software includes drivers, GUI and also the operating system. This kind of platform software is packaged along with the computer but it is possible to change the platform software of the personal computers.

- **Application Software**
- This kind of software usually doesn’t come bundled up with the computer but they have to be bought individually. These are the ones, which are used to play video games, office suites and also include other software like word processing software, video games, office suites, word processing software, presentation software, multimedia software etc. Even systems software can be thought of as application programs. In some cases, application software are pre-packaged with the computer but still, they are run as separate programs. Even systems software can be thought of as application programs.
- **User written software: This software are created by the users to suit their specific needs. They can also be bundled within default application software and actually if they are deftly packed, they are indistinguishable from the original bundle of software. E-mail filters, spreadsheet templates etc. are all user written software.**

Importance of quality of software: It is rather quite significant to have good quality software because many important jobs are dependent on the software especially in MS office, windows, Linux etc., we cannot afford to have bugged software. Faulty software can damage the system and also crash it. Moreover, it may generate undesirable and unforeseen responses. Quality in software can be defined in terms of code characteristics. It is actually the conformation of all characteristics required from a good professional software e.g. functional and performance needs (speed, storage area needed, security, network friendliness etc. and document development features. Nevertheless, its quite difficult to measure the quality of as software e.g. we can objectively measure speed, power consumption but security issues are very difficult to assess.

Features of Software

The software are basically prepared to deliver a particular facility. These facilities include the types of output generated by the software e.g.

- Numerical result
- Screen shots
- Web pages etc. and the features of the software to deliver a particular kind of service are deliberated in terms of Functionality, execution, movability and compactness.

- Speed: If the application is asked to perform a function, it takes a little bit of time before delivering the service asked for. The time taken in performing the function is sum total of three different things i.e.
- System time
- CPU time
- Network latency
- Total Wall time or real elapsed time

So, in order to measure develop a software on lines of a good speed software, the developer must take into account all the above mentioned times.

- **Space utilization of the software:** The RAM or disk space utilized by the application is of prime importance. Software, which stimulates CPU cache miss, can also influence the speed of the software. With the starting of computer, many software also get started, some of them are not actually required at that time. Even this slows down the computer and it needs to be managed properly to keep vital resources of the computer free for usage. Here, data design of the software is very important as a poor design results in slowing down of the system.
- **Network usage:** Network usage can be defined as the network traffic at particular time as compared to the total amount of traffic that can be supported by the network. Bandwidth usage involves all incoming (data downloaded from net) and outgoing (data sent out from the computer in form of emails. etc.) Software should be so designed that it efficiently uses the bandwidth.

- **Durability of the software:** Good quality software doesn’t need to be repaired very often. It is very troublesome at the users end to fix the software again and again. The code used to write the software should not be fragile otherwise the patchwork has to be done again and again.
- **Strength of the software**: Good quality software possesses good tolerance towards system failure, unresponsive third party resources. In case, the quality of the software is not good it becomes very prone to freezing or crashing. So, software should be tolerant enough to severe and testing conditions.
- **Simplicity for user**: Actually, ease to use software is not very objective. It is a relative and matter, which is difficult to calculate. The various parameters upon which the quality of a software can be determined is how clearly it conveys the errors, how it recovers after failure etc.

- **Reproducibility:** It is additionally recognized as determinism or repeatability. It is a parameter that takes into account the repeatability of results i.e. whether the software gives the same result time after time. There are numerous reasons responsible for non-reproducible results, the chief of which is a bad data model design.
- **Issue of compatibility:** This issue addresses the compatibility between new and older versions of the software. Further, good software should be so designed that shifting of data from old to new should not take a lot of time/expensive and it should be cost effective.
- **Safekeeping**: Good software should be secure and should possess restricted access if so desired. The data, which is processed by the software, should not be compromised. Actually, this aspect is of utmost importance and anybody who wants to keep their data confidential must have secure software.

- **Consumption of power:** As most of the current day smart phones are having such application software which are used quite often by the user, it is but obvious that the software must not use too much of battery (power) so that battery remains charged for a longer time.
- **Code coverage:** Good quality software has higher code coverage. Code coverage is actually a parameter, which illustrates the degree to which the source code of the program has been tested. Software which has higher code coverage means that it has been sufficiently tested and has a lesser probability of containing software bugs.
- **Testability:** Good software is designed to possess high level of testability. High testability needs an inter-dependence between architecture as well as testing methods used. An inadequate software with poor testability needs to be redesigned which is an extremely costly affair. So, at the initial step only it is justifiable to invest in software architecture and development methodologies to prevent repairs and redesigns.

- **Portability:** Good quality of software possesses portability i.e. they possess the ability to run on different types of computers as well as operating systems. including small capacity computers, mobile phones, various web browsers etc.
- **Safety of thread:** Good software should not get into a logjam. Improper thread safety of software leads to live locks, resource famishing and deadlocks. A code should be so written that it may be partly executed by thread, executed by the same thread and also by another thread.
- **Compactness:** Good quality software needs to have smaller binary size and also should possess rapid compilation. If software is smaller in size, it will have fewer bugs because no. of bugs remains persistent with respect to size of the code.

- **Maintenance:** Good quality software is easy to debug, faster to fix, easier to understand. Maintenance of software is improved with good testability and malleable design. Software maintenance is a very important aspect and it is also subjective in quantification.
- **Documentation:** Many users are of a view that side by side separate documentation written in English language is pretty essential. On the other hand there are people who testify that at least one fourth of the code should be in form of commands. A third party believes that code in itself is the most fit manifestation of documentation.
- **Software should be legible:** Legible means readable. Readability of the code is very important. The good software should be having a unified style of code. Being legible means that a software should be easily read by the developer even if its written by another developer.

- **Software should be saleable:** All the good software should have the ability to easily extend a certain feature, enlarge or add novel features, additional chores etc. In fact, good software is developed keeping the potential prospects needs in mind. The quality of the software is actually the reaction of the user to it, a user friendly software which gives good results and positive feedback from user is subjectively a good quality software. Nevertheless, it must kept in mind that the quality of software is actually the result of good ground work and focused on quality parameters like stability, compactness, portability, network usage, code coverage, testability, flexibility, security, thread security, maintainability, legibility, scalability, documentation.

Factors controlling the choice/selection of particular software

- Whether the need is of a single user software or multi user software in a particular place or multi user in physically diversified areas
- Whether the software proposes to give certain tools that can be used for supervising by the administration
- Whether it offers security or not. Price Tag of the software: The price of the software includes all the prices i.e. the initial cost, cost of installation, cost of up gradation, cost of maintenance and also the cost involved in buying different versions and single/multi user license.
- Competence of the software: The issue of competence of software is very crucial. It has to be taken into account that the software that being bought today, whether it is also fit and competence to meet the potential needs of tomorrow.
- Software compatibility: The compatibility issues of the software are also of prime importance because the software needs to be compatible with the prevailing hardware and software combinations.

Licensing and authorization issues of the software

- All the software, which is developed by various companies, should be licensed because a lot of effort, labor, expertise and skill go into the development of software. The software must not be copied, pirated and sold without the permission of the developing company. So, licensing is a major issue of concern for the current needs.

Legal Entitlements over the software

- All the laws that work to provide the intellectual protection rights to other intellectual properties have to be extendable to the software also. Actually, there needs to be a legal entitlement over the software because it has been prepared but inputs of human mind and intellect, innovation and creativity. In fact, the Software Publishers association imposes the copyright rules and commandments in the companies, corporations etc via the conventional recommendations. According to these conventions, if the information system managers do not possess the evidence of buying the software, they have no right to continue possessing the software, else they must buy new license, failing which, they stand liable to punishment. Further, they must pay the fee for renewal/up gradation of the license. By definition, a license is actually an entitlement approved by the government to carry out an activity, which is otherwise illegal. Software Publishers association keeps a check on the companies to invigilate the appropriate utilization of the license. If a company is found guilty of improper utilization of the license, it is accountable and liable to substantial fine. These days the information system managers are thoroughly checking records and maintaining records for the correct licenses of their software.

- Many people unmindfully copy the original software but little do they realize that doing so is criminal and punishable by law. Furthermore, the cost of piracy runs in millions in terms of money, time and security. Many people who use pirated software in the hope of saving a small amount of initial investment fall prey to the malware and security threats associated with the pirated or copied software. A research has shown that more than sixty percent of companies who have used forged software have faced huge security threats. Moreover, the same research showed that about half of the consumers faced the problem of slow PC and had to remove the software. In addition the users of counterfeited software were very much concerned about data loss. If small companies indulge in such a piracy, they are more gullible to loss because large companies are rich enough to face huge losses but smaller companies are on he risk of getting completely ruined due to incurrence of losses.
- Ethically speaking also, the cause of innovation and intellect is also defeated and it is ethically incorrect to do so. However, as the number of computers and computer companies are on the rise, the business of fake software is also in the rise. This in turn has caused a number of tracking software companies to grow.