

DIPLOMA WALLAH

(Your One Stop Hub For Diploma Resources)

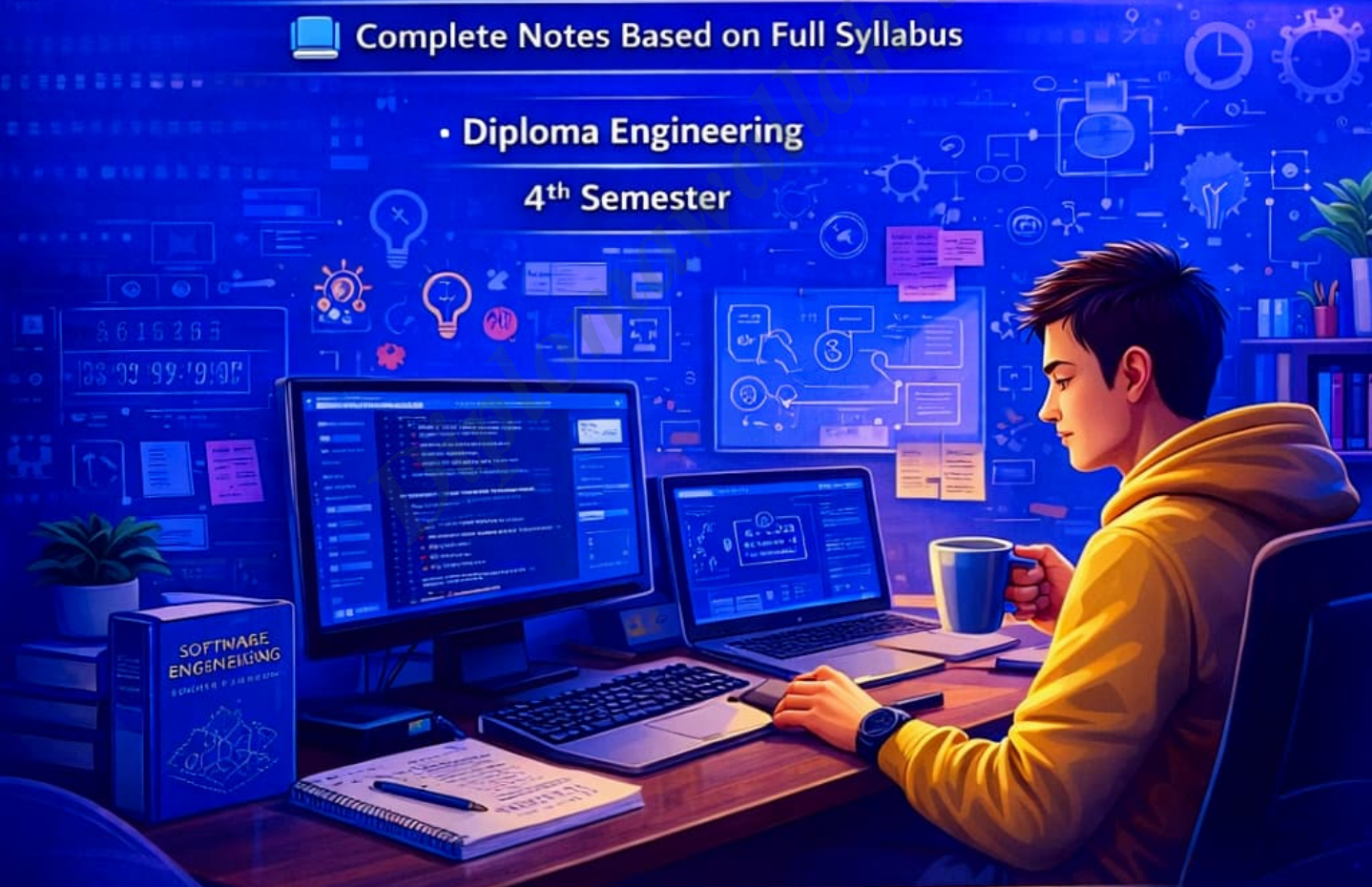


SOFTWARE ENGINEERING PRINCIPLES AND PRACTICES

 Complete Notes Based on Full Syllabus

• Diploma Engineering

4th Semester



© Diploma Wallah. All Rights Reserved

Unauthorized sharing/selling is strictly prohibited

www.diplomawallah.in

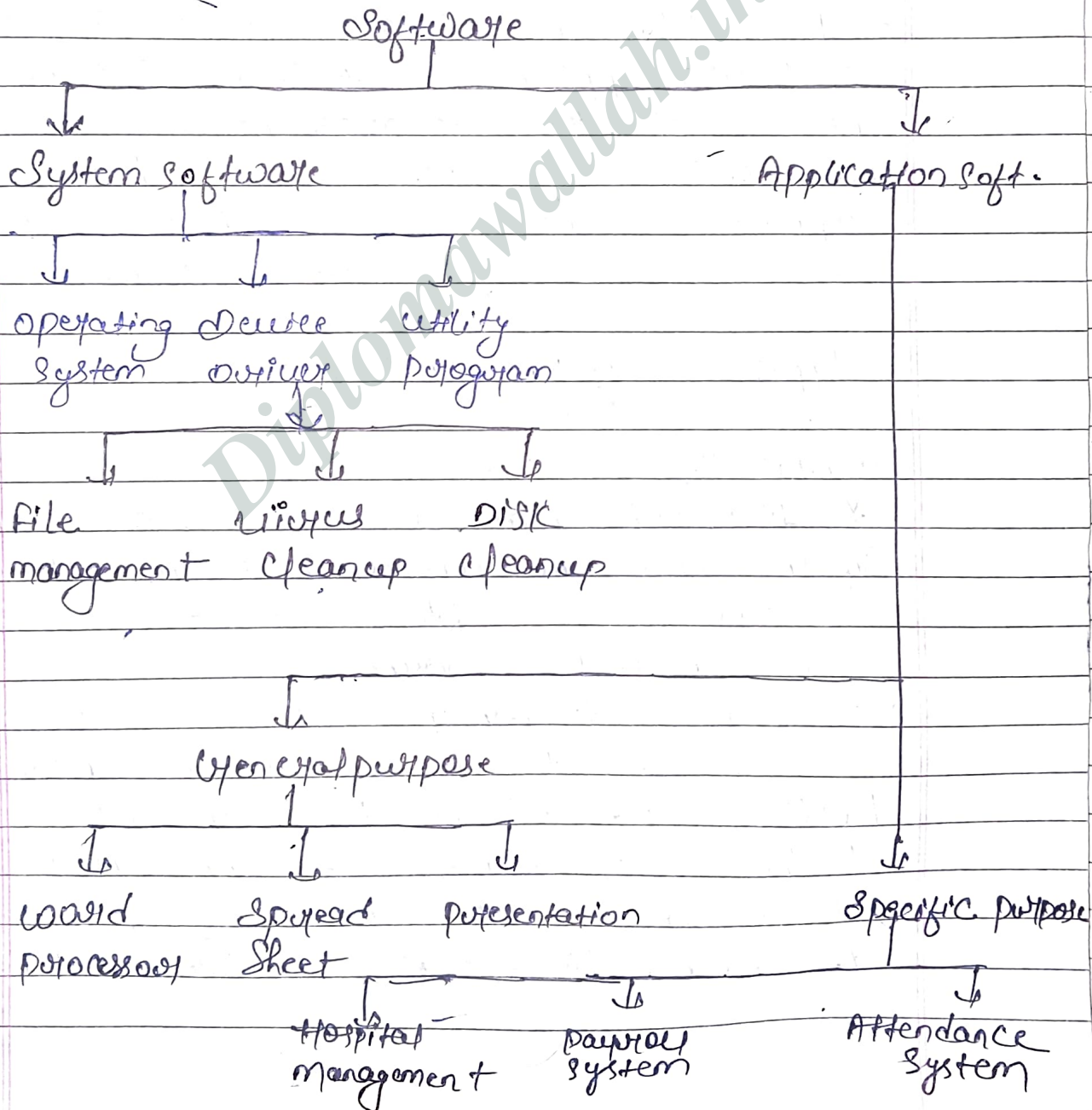
Notes prepared by Sangam

Software (Unit-01)

It is a set of program or

It is basically a set of instructions or commands that tell a computer to do a specific task that serves its user.

Examples:- ms word, ms-excel, powerpoint and web browser.



System Software:- It is a software that directly operates the computer hardware and provide the basic functionality to the user as well as to the other software to operate smoothly.

- System software basically controls a computer's internal functionality and also hardware such as monitor, printers and storage devices etc.
- It is like an interface b/w hardware and user application, it helps to communicate with each other because hardware understand machine language (i.e. 1 or 0) where user applications are work in human-readable language like English, Hindi, German etc.

Types of System Software

1. **Operating System :-**

* It is the main program of a computer system when the computer system on it is the first software that loads into the computer memory.

* Basically, it manages all the resources such as computer memory, CPU, printer, hardisk etc. and provides an interface to the user, which help the user to interact with the computer system.

Ex- windows, linux, macOS, Android, ios etc.

Device driver is a special type of OS that acts as a translator b/w computer hardware.

→ It acts like middleman that convert the OS commands into hardware understand, and vice versa.

2. Device driver: - A device driver is a program or software that controls device and help that device to perform its functions.

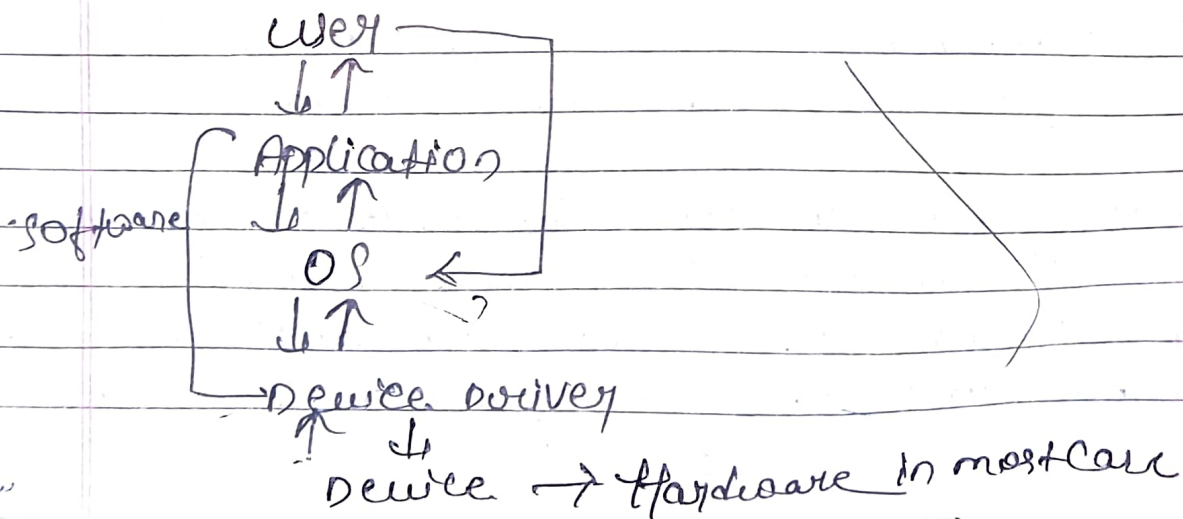
* Every devices like a printer, mouse, etc need a driver to connect with the computer system. So, when we connect a new device with our computer system we need to install the driver of that device so that your operating system knows how to control or manage that device.

3. Utility Software: -

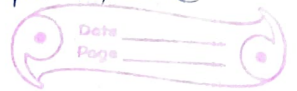
This type of application software is used to support the computer infrastructure. It is designed to analyze, configure, optimize and maintain the system and take care of its requirement as well.

Example: - Antivirus, disk fragments, memory tester, disk repair, disk cleaners, registry cleaners, disk space analyzers etc.

→ Device driver



A branch of engineering where software are made and managed.



~~* Needs of~~

Software Engineering :-

The IEEE defines Software engineering as the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software.

* Systematic :- Following a structured and organised method not random or ad-hoc. (Planned structure work)

* Disciplined :- Sticking to best practices, rules, and standards.

* Quantifiable :- can be measured (like time, cost, performance, quality).

* Development :- Creating the software (designing, coding, testing)

* Operation :- Running using the software in real environments.

* Maintenance :- Fixing bugs, adding features, updating software over time.

In simple words, Software engineering is about building and maintaining software in a professional, structured, measurable and quality-assured way.

* Need of Software Engineering

- Software engineering is a technique through which we can develop or create software for computer system or any electronic device.
- It is a systematic scientific and disciplined approach to the development, functioning, and maintenance of software.
- Software engineering builds the software and application by using designing and programming language.
- ~~How~~ In order to create complex software, we need to use software engineering technique as well as reduce the complexity we should use abstraction and decomposition.
Decomposition: Breakdown of the software in a number of module, where each module procedure as well defines the independent task.
- Handling big project: - A corporation must use a software engineering methodology in order to handle large project without any issue.
- To manage the cost: - Software engineering programme plan everything and reduce all these things that are not required.

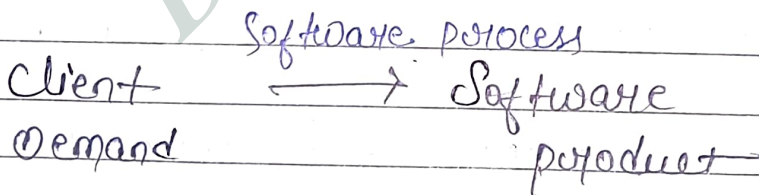
◦ Reduce time: - It will save a lot of time if you are developing software using a software engineering technique.

◦ Reliable Software: -

It is the company's responsibility to deliver software product on schedule and address any defect that may exist.

◦ Effectiveness: - Effectiveness result from things being created in accordance with the standards (work done in a specific systematic time.)

◦ Reduce Complexity: - large challenge are broken down into smaller one and solved one at a time in S/W engineering



↓ approach

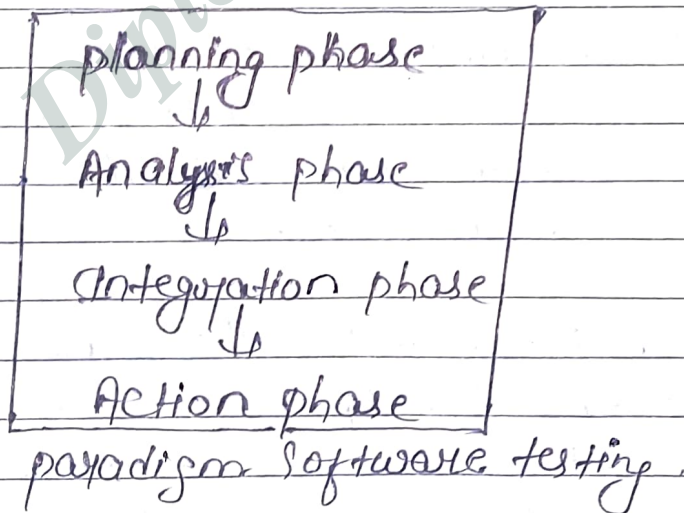
Software paradigm :-

Software paradigm refers to a paradigm. A paradigm comes from the Greek word means "A way of thinking, a model, or a standard method of doing something."

In terms of software :-

A standard approach or methodology used to build and organize software system.

* Software paradigm refers to method and steps which are taken while designing the software. Software is considered to be a collection of executable programming code, associated libraries and documentation. Software development paradigm is also known as software engineering.



There are several software paradigm including:-

Imperative paradigm:-

It is a programming approach where code is written as a sequence of instructions that tell the computer how to perform tasks step by step.

Object-Oriented paradigm:-

This paradigm is based on the idea of object which are self-contained unit that contains both data and behaviour.

It is often used in language such as Java, C++ and python.

• **Functional paradigm:-** This paradigm is based on the idea of that a program is a set of mathematical functions that transform input into outputs.

• **Logical paradigm:-** This paradigm is based on the idea of that is a program is a set of logical statements that can be used to represent new information.
(It is) often used in language such as Prolog and Mercury.

o The Software development life cycle (SDLC) is a process that software developers used to plan, design, develop, test, deploy and maintain software system.

The most common SDLC models includes: -

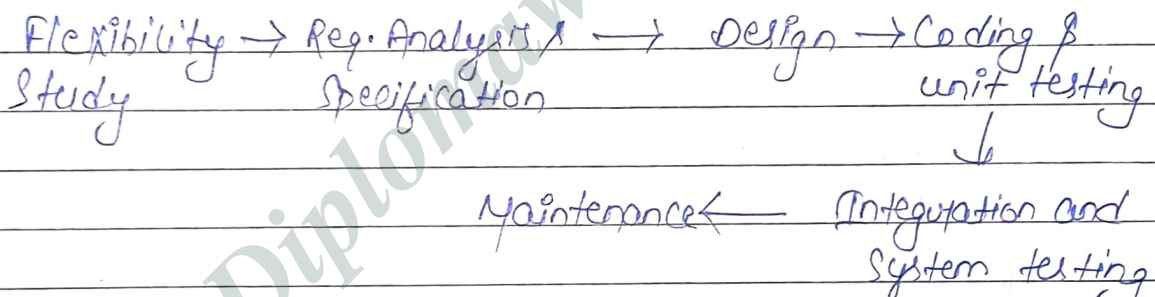
o Waterfall models: -

· A linear step-by-step approach where the next phase begins only after the previous one is completed.

Features: -

- Simple and easy to understand.

- Best for project with well defined req.



* main disadvantage is that no changes allowed once the process is done.

o Agile model: - This model is based on the idea that software development is an iterative process with small. It focus on continuous development and feedback.

Software product:-

It refers to computer programs and related documentation that are created to meet specific user needs.

They are generally divided into two main types:-

1. Generic Software product
2. Customised Software product.

1. Generic Software product:

It is pre-designed, mass-market software product developed by a software company for general use, not for any specific user.

Characteristics

Pre-built:- made before knowing who will use it.

Multi-user:- Designed for a broad range of users.

Configurable:- offers some settings you can change (but not deep customization)

off-the-shelf:- you can buy and start using immediately.

Shared Cost:- Development cost is shared across many customers.

Example:- ms office (word, excel), Adobe photoshop, VLC media player, Google Chrome, Zoom/Skype etc.

Advantages

1. Cost-effectiveness:- Low price, because it is sold to many users.
2. Quick to use:- Ready for download or install.
3. Reliable & Tested:- Used by millions - few bugs.
4. Help Available:- documentation, tutorial, forums.
5. updates:- Regular updates from the vendor.

Disadvantage

1. Limited customization:- we can't change core features.
2. Extra features:- includes things you might never use.
3. Not specific:- may not match your business exactly.
4. Dependency:- you rely on the vendor for changes.



11

Customized Software product:—

A customized software product is specially developed for a particular user, company or organization based on their specific requirements.

Also known as Bespoke software.

Customised software = Software made on order, just like a tailored suit is made to fit a specific person.

Characteristics

- i) Tailored:— Developed after understanding client needs.
- ii) private use:— only used by one client or organisation.
- iii) Scalability:— Can be expanded or modify easily in the future as the organisation grows.
- iv) Security:— Security features are added based on the client internal data protection needs.

Advantage

Tailored to your needs.

Better efficiency

Easy integration

Disadvantage

High cost

Time-consuming

Maintenance required

Initial training may be needed.

Require good communication.

Characteristics of Good Software

A software product is considered as good software by what it offers and how well it can be used. The factors that decide the software properties are divided into three categories: Operational, Transitional and maintenance.

1. Operational:- In Operational Categories, the factors that decide the software performance in operation. It can be measured on:
Budget, usability, efficiency, compactness, functionality, dependability, security, safety.

2. Transitional:-

When the software is moved from one platform to another, the factors deciding the software quality, portability, interoperability, reusability, adaptability.

3. Maintenance:-

In this category, all factors are included that describe about how well a software has the capabilities to maintain itself in the ever-changing environment.

modularity, maintainability, flexibility, scalability.

Challenges in Software projects and factors affecting success.

Developing a software project is not just about writing code - It's about managing people, planning, technology, communication, risks, and many more.

Some key challenges are:-

1. poor Requirement understanding:- If the team does not clearly understand what the client wants they will build the wrong product.

Impact:- wasted time, money and efforts.

2. Lack of communication:- When the team, clients or departments don't talk clearly or regularly.

Example:- A developer make a change that the client did not approve.

Impact:- Confusion, delays and misunderstanding.

3. Unrealistic Time Deadlines:- Giving very short or impractical time for project completion.

Impact:- poor quality, stressed developers, and missed deadlines.

4. Inexperienced Team:- Developers or testers who don't have enough knowledge or skills.

Impact: more bugs, errors and failed projects.

5. Budget Constraints:- Not enough money to complete the project properly.

example:- not having funds for testing or better hardware.

Impact: incomplete feature, low performance, and low quality.

6. Changing Requirement (Scope creep):- When the client keeps asking for new features during the development.

example:- Adding chat, payment and cups, after the project is halfway done.

Impact:- Delay, increased cost and confusion.

7. Technology Issue:- Using outdated or wrong technologies.

example:- using language that doesn't support the needed features.

Impact: Compatibility issue, bugs and slow down.

8. Lack of user involvement: user are not involved during planning or testing.

example:- making a student app without asking student what they needed.

Impact:- product doesn't meet user needs.

* Software process:-

Software process in software engineering refer to the methods and techniques used to develop and maintain software. Some example of software process includes:-

Waterfall:- Linear and sequential. Each phase must be completed before moving to the next. Once a step is done you cannot go back easily.

Phase → Requirement → Design → Implementation → Testing → Deployment → Maintenance.

* **Agile:-** Iterative and flexible development. The software is built and delivered in small parts (called sprints). Customer feedback is taken in every cycle.

* **Scrum:-** Focuses on teamwork and sprints. Daily short meetings are held (called daily stand up).

* **DevOps:-** A set of practices that aims to improve collaboration and communication b/w development and operation teams, with an emphasis on automating the software delivery process.

Software Component are:-

1. Code:- The instruction that a computer executes in order to perform a specific task or set of tasks.
2. Data:- The information that the software uses or manipulates.
3. user interface:- By which the user interacts with the software, such as buttons, menus, and text fields.
4. Libraries:- Pre-written code that can be reused by the software to perform common tasks.
5. Documentation:- Information that explains how to use and maintain the software, such as user manuals, and technical guides.
6. Configuration files:- Files that contain settings and parameters that are used to configure the software to run in a specific environment.
7. Build and deployment:- Tools that are used to build packages, and deploy the software to different environments.
8. metadata:- Information about the software, such as version numbers, authors, and copyright information.

Key process Activities

There four basic key process activities are:-

1. Software Specification:-

In this process, detailed description of a software system to be developed with its functional and non-functional requirements.

2. Software Development:-

In this process, 'designing, programming, documenting, testing, and bug fixing' is done.

3. Software validation:-

In this process, evaluation software product is done to ensure that the software meets the business requirements as well as the end user needs.

4. Software evolution:-

It is a process of developing software initially then timely updating it for various reasons.

Diploma Wallah

Pengam

Sharing/selling not allowed.