SOFTWARE

Ampuyas, Marc Jay M.
Sevilla, Angelica S.

Evolving Role

- **“Software is both a product and a vehicle that delivers a product.”**
- **Today, a huge software industry has become a dominant factor in the economies of the industrialized world. Teams of software specialists, each focusing on one part of the technology required to deliver a complex application, have replaced the lone programmer of an earlier era.**

Evolving Role

- **The role of computer software has undergone significant change over the last half-century. Dramatic improvements in hardware performance, profound changes in computing architectures, vast increases in memory and storage capacity, and a wide variety of exotic input and output options, have all precipitated more sophisticated and complex computer-based systems.**

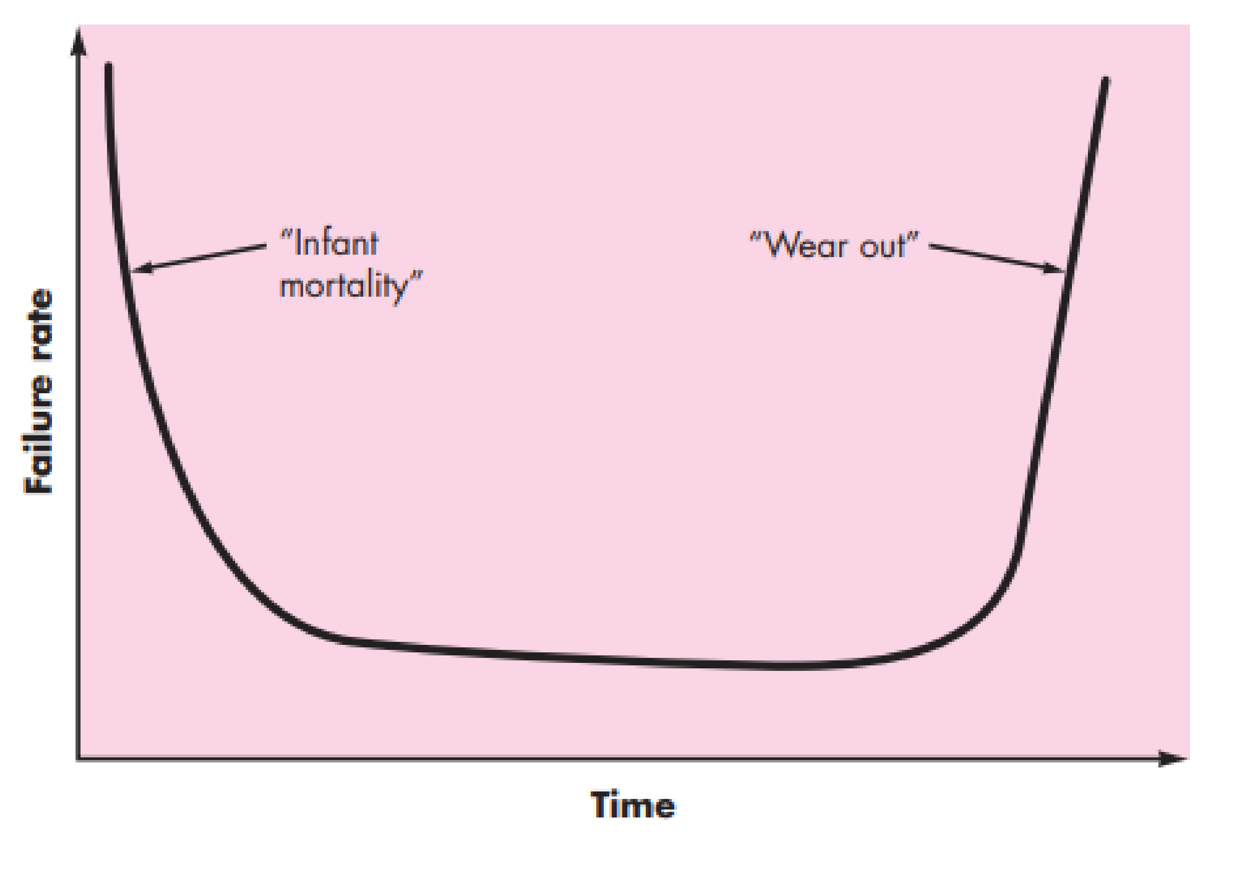
WHAT IS SOFTWARE?

- Software is a collection of instructions, data, or computer programs that are used to run machines and carry out particular activities.
- Data structures that enable the programs to adequately manipulate information.
- Software is a logical rather than a physical system element.

CHARACTERISTICS

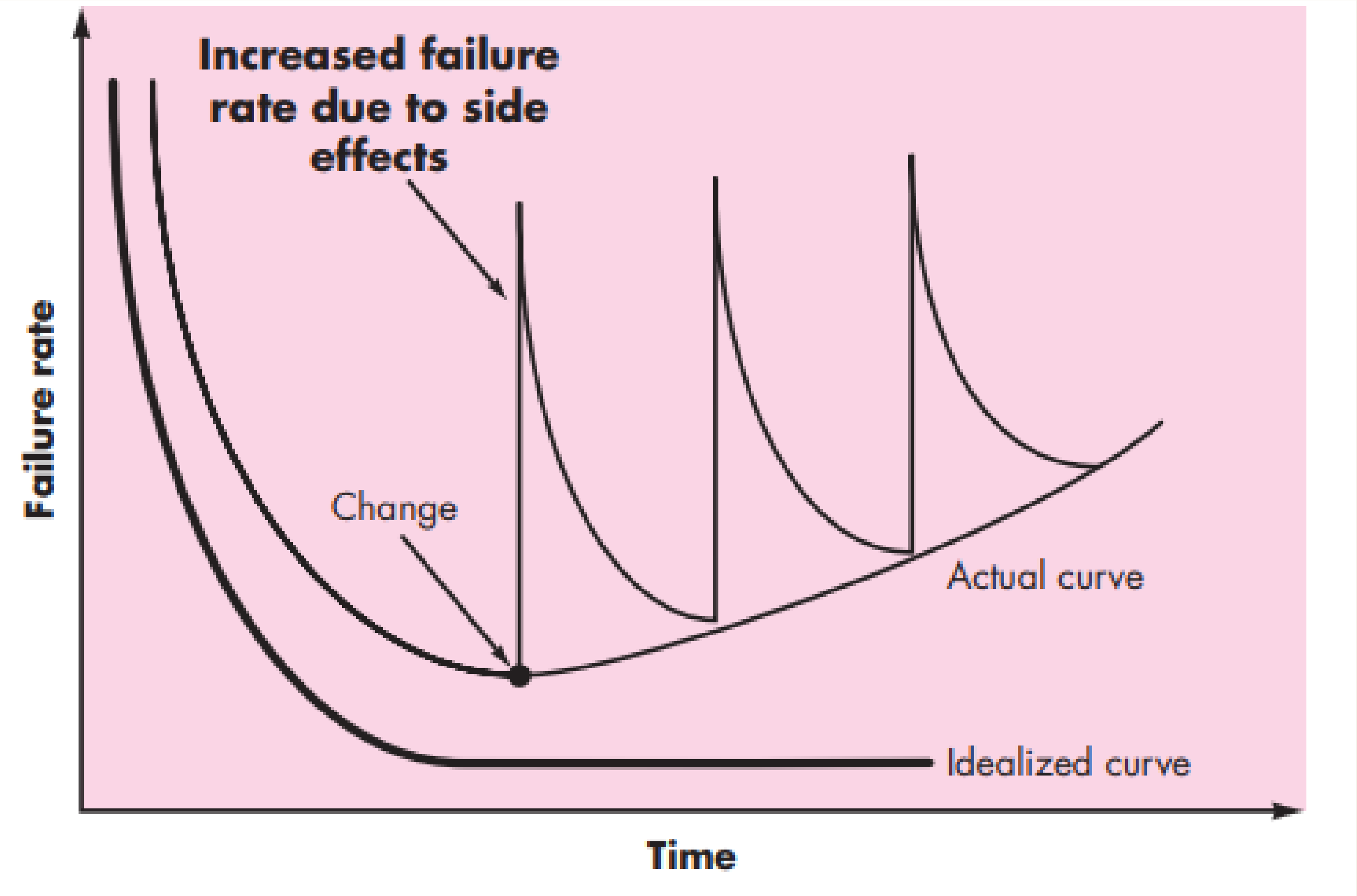
- Software is engineered, not manufactured.

- Software doesn’t wear out, but it does deteriorate.



CHARACTERISTICS

- **Software engineering methods strive to reduce the magnitude of the spikes and the slope of the actual curve in the figure.**



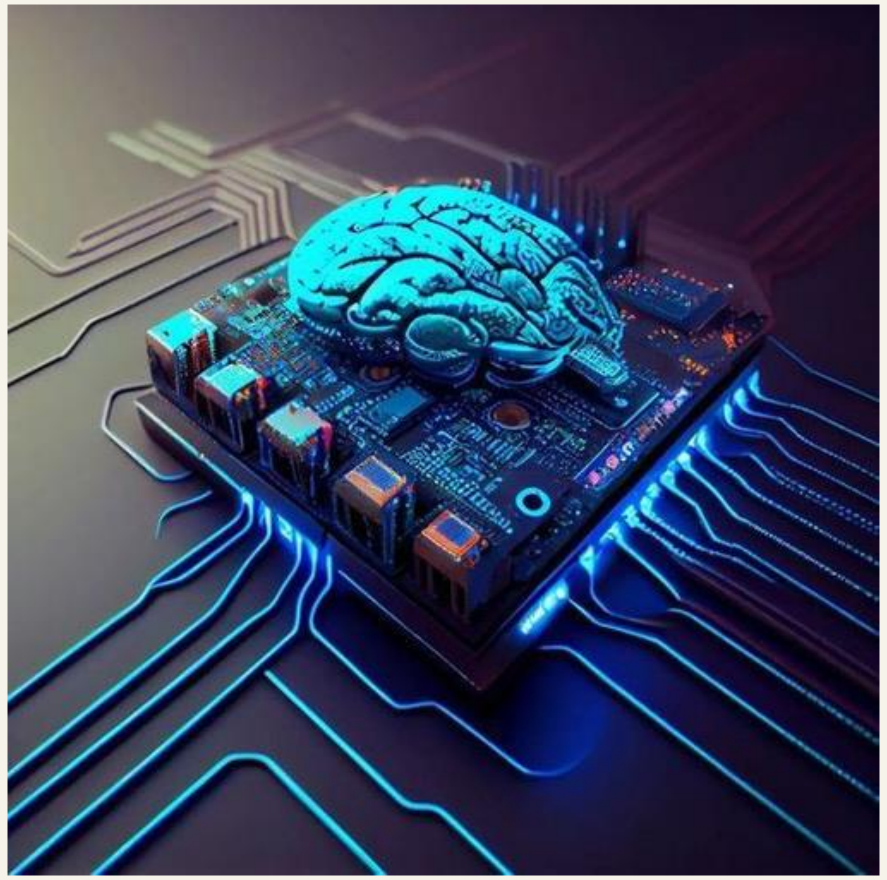
SOFTWARE APPLICATION DOMAINS

- **System software** — a collection of programs written to service other programs. Some system software (e.g., compilers, editors, and file management utilities) processes complex, but determinate,4 information structures.

Example:

1. **Operating Systems(Windows, Mac OS & Linux)**
2. **File Management Utilities**
3. **Device Drivers**
4. **Compilers**

- **Software Applications** — in this area process business or technical data in a way that facilitates business operations or management/technical decision making. In addition to conventional data processing applications, application software is used to control business functions in real time (e.g., point-of-sale transaction processing, real-time manufacturing process control)



Example:

1. **Microsoft Word**
2. **Netflix**
3. **Facebook**
4. **Firefox**

- **Engineering/scientific software**—has been characterized by “number crunching” algorithms. Applications range from astronomy to volcanology, from automotive stress analysis to space shuttle orbital dynamics, and from molecular biology to automated manufacturing.

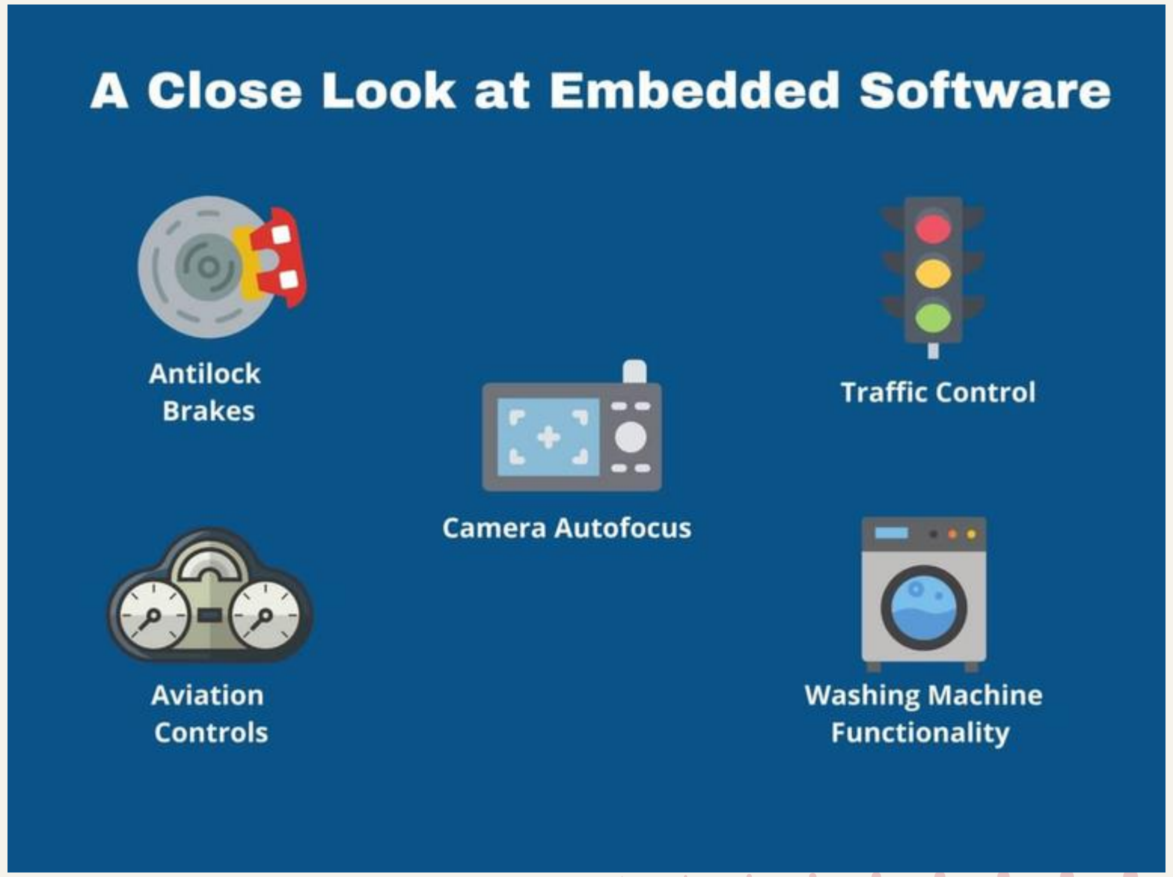
Example:

1. **MATLAB**
2. **AutoCAD**
3. **ANSYS**
4. **Simulink**

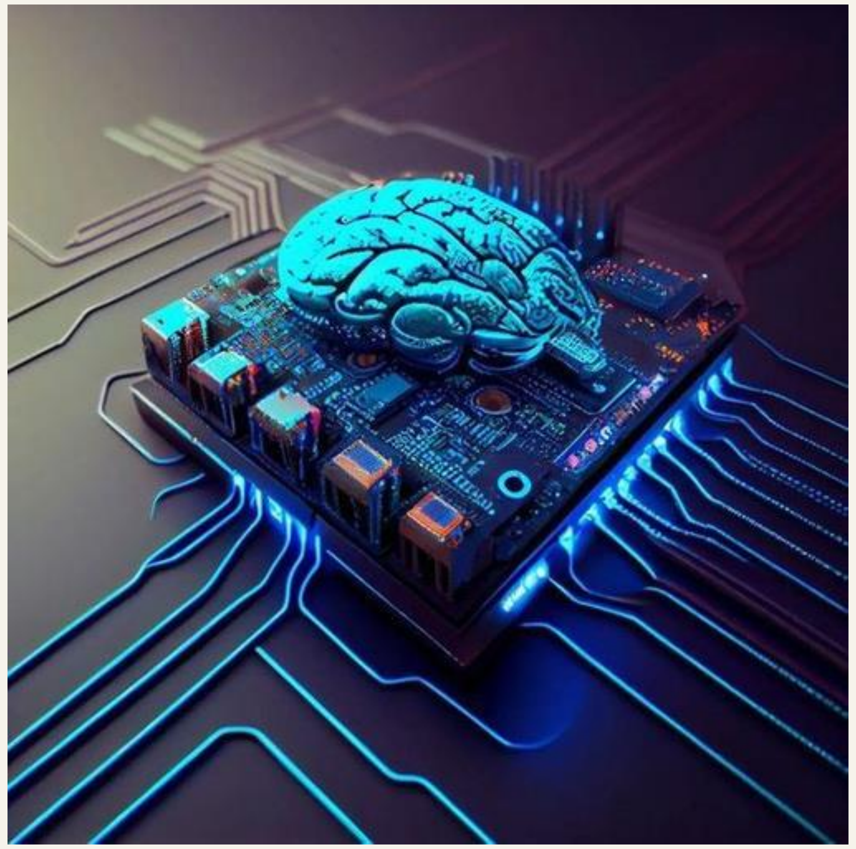
- **Embedded software**—resides within a product or system and is used to implement and control features and functions for the end user and for the system itself. Embedded software can perform limited and esoteric functions (e.g., key pad control for a microwave oven) or provide significant function and control capability (e.g., digital functions in an automobile such as fuel control, dashboard displays, and braking systems).

Example:

1. **Stand Alone(** Microwave Ovens, Washing machines, & Video game consoles**)**
2. **Networked (**ATMs, Home security systems,& Card swipe machines)
3. **Mobile (**Digital cameras, Mobile phones, Smart watch, Fitness tracker**)**



- **Product-line software**—designed to provide a specific capability for use by many different customers. Product-line software can focus on a limited and esoteric marketplace (e.g., inventory control products) or address mass consumer markets (e.g., word processing, spreadsheets, computer graphics, multimedia, entertainment, database management, and personal and business financial applications).



Example:

1. **Word Processing Software**
2. **Spreadsheet Software**
3. **Database Management Systems**
4. **Entertainment Software**

- **Web applications**—called “WebApps,” this network-centric software category spans a wide array of applications. . In their simplest form, WebApps can be little more than a set of linked hypertext files that present information using text and limited graphics.

Example:

1. **Simple WebApps(** blog or a news website **)**
2. **Advanced WebApps in Web 2.0**

- **Social media platforms(** Facebook or Twitter)
- **Google Docs**
- **Online banking**

cmarix
Envision · Leverage · Infinite
7 Top-Rated Web Application Examples and Beyond
NETFLIX

- **Artificial intelligence software**—makes use of nonnumerical algorithms to solve complex problems that are not amenable to computation or straightforward analysis.

Example:

1. **Robotics**
2. **Expert Systems**
3. **Theorem Proving**
4. **Artificial Neural Networks**

- **Open-world computing**—the rapid growth of wireless networking may soon lead to true pervasive, distributed computing. The challenge for software engineers will be to develop systems and application software that will allow mobile devices, personal computers, and enterprise systems to communicate across vast networks.

Example:

1. **Smart Homes**
2. **Google Maps on Mobile**
3. **Self-Driving Cars**
4. **Wearable Health Devices**

- **Netsourcing—**the World Wide Web is rapidly becoming a computing engine as well as a content provider. The challenge for software engineers is to architect simple (e.g., personal financial planning) and sophisticated applications that provide a benefit to targeted end-user markets worldwide.

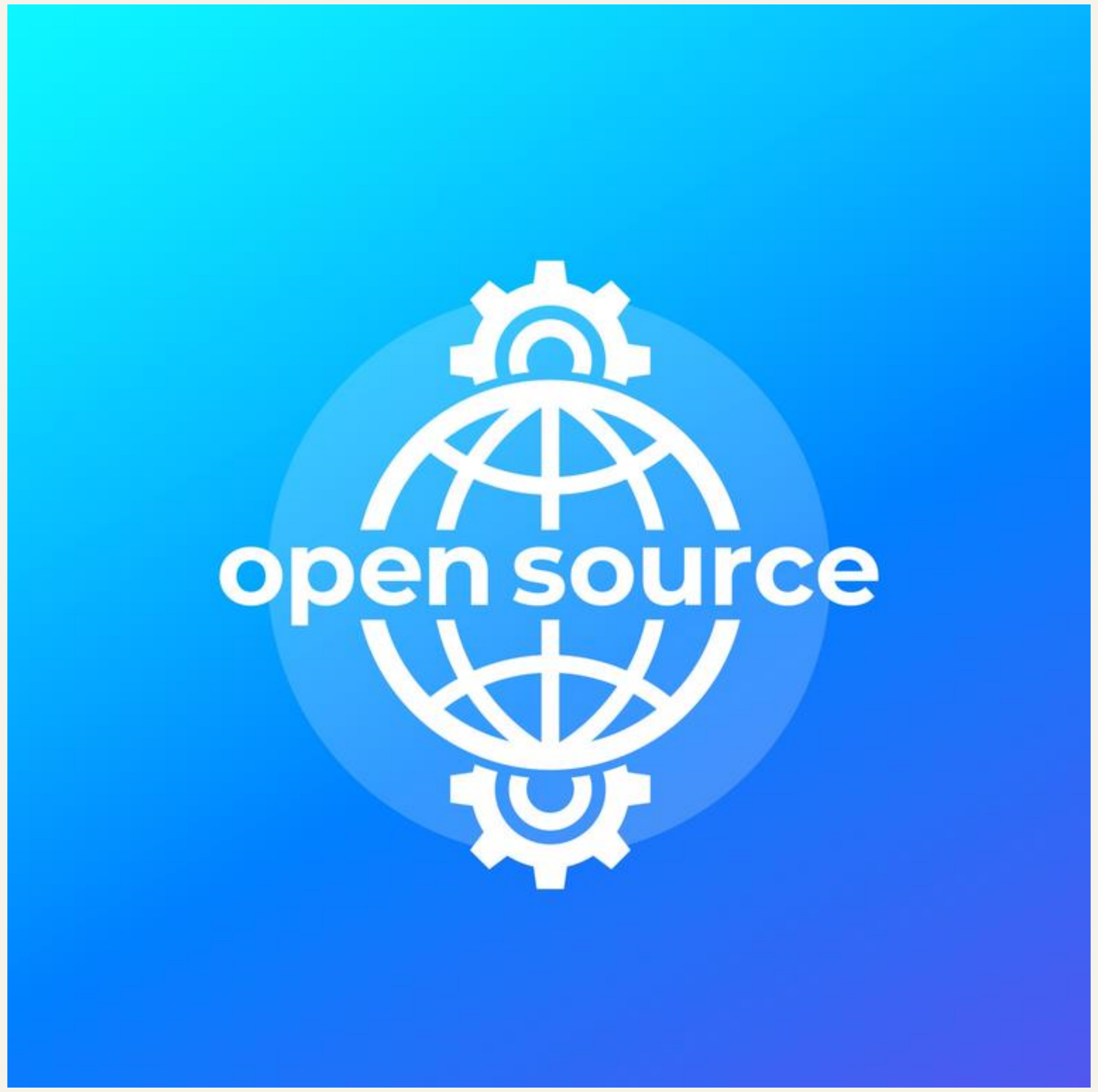
Example:

1. **Google Docs**
2. **Dropbox**
3. **Salesforce**
4. **Netflix**

- **Open source**—a growing trend that results in distribution of source code for systems applications (e.g., operating systems, database, and development environments) so that many people can contribute to its development.

Example:

1. **Linux**
2. **Mozilla Firefox**
3. **MySQL**
4. **Git**



LEGACY SOFTWARE

- Legacy software refers to old computer programs that have been used for a long time—sometimes decades—but are still important for businesses, industries, or government operations. These programs were created using older technologies but have been updated over time to keep up with changing needs.

Example:

1. Banking Systems (COBOL)
2. Airline Reservation Systems
3. Retail POS (Point of Sale) Systems
4. Government Systems

Legacy Software

If the legacy software meets the needs of its users and runs reliably, it isn't broken and does not need to be fixed. However, as time passes, legacy systems often evolve for one or more of the following reasons:

- The software must be adapted to meet the needs of new computing environments or technology.
- The software must be enhanced to implement new business requirements.
- The software must be extended to make it interoperable with other more modern systems or databases.
- The software must be re-architected to make it viable within a network environment.

UNIQUE NATURE OF WEBAPPS

In the early days of the World Wide Web (1990 to 1995), websites consisted of little more than a set of linked hypertext files that presented information using text and limited graphics.

Today, WebApps have evolved into sophisticated computing tools that not only provide stand-alone function to the end user, but also have been integrated with corporate databases and business applications due to the development of HTML, JAVA, xml etc.

1.Network intensiveness.

- A WebApp resides on a network and must serve the needs of a diverse community of clients.

2. Concurrency : [Operation at the same time]

- A large number of users may access the WebApp at one time. In many cases, the patterns of usage among end users will vary greatly.

3. Unpredictable load :

- The number of users of the WebApp may vary by orders of magnitude from day to day. One hundred users may show up on Monday; 10,000 may use the system on Thursday.

4. Performance :

- If a WebApp user must wait too long (for access, for server side processing, for client-side formatting and display), he or she may decide to go elsewhere.

5. Availability :

- Although expectation of 100 percent availability is unreasonable, users of popular WebApps often demand access on a 24/7/365 basis.

6. Data driven :

- The primary function of many WebApps is to use hypermedia to present text, graphics, audio, and video content to the end user.

7. Content sensitive:

- The quality and artistic nature of content remains an important

Determinant of the quality of a WebApp.

8. Continuous evolution:

- Unlike conventional application software that evolves over a series of planned, chronologically spaced releases, Web applications evolve continuously.

9. Immediacy:

- Although immediacy—the compelling (forceful) need to get software to market quickly—is a characteristic of many application domains, WebApps often exhibit a time-to-market that can be a matter of a few days or weeks.

10. Security:

- Because WebApps are available via network access, it is difficult, if not impossible, to limit the population of end users who may access the application.

11. Aesthetics : [Artistic / Visual]

- An undeniable part of the appeal of a WebApp is its look and feel. When an application has been designed to market or sell products or ideas, aesthetic may have as much to do with success as technical design.

REFERENCES

- Roger S. Pressman, Roger S. Maxim, Bruce R. Dr (2019), Solfware Engineening. A Practitioner's Approach (th Edition) Mograw-Hill Education
- Written by Indeed Editorial Team Updated August 16, 2024 from https://www.indeed.com/career-advice/career-development/what-is-software-product-line
- Software product line Examples by: Prof. Dr. Fazal Rehman Shamil | Last updated: December 30, 2023 from https://t4tutorials.com/software-product-line-examples/
- Software Engineering, Wednesday, November 23, 2016 from https://softwareengineeringmca.blogspot.com/

THANK YOU

Ampuyas, Marc Jay M.
Sevilla, Angelica S.