Discussion-1 Seven Broad Categories of Computer Software

Discussion Topic:

For this discussion, present and evaluate a commonly used definition of software. Reflect on these questions as you prepare your answer:

Is the definition adequate?
What is missing?
Why is it difficult to define software?
Does this definition keep in mind best practices of software development?

The textbook and this module's lecture provide seven broad categories of computer software. Do you think the same approach to software engineering can be applied for each? Explain your answer.

My Post:

Hello Class,

One definition describes software as having three core components:

1. Programs, which are the instructions that tell a computer what to do (Pressman and Maxim, 2020)
2. Data structures, which organize the information the programs use
3. Documentation, which provides information on how to use and maintain the software, as well as explaining its business logic.

I think that this description of software, as a set of three core components, is a somewhat adequate definition of what software is. As it includes not just “code”, the program, but also the data structures dictating which information the programs use, and the documentation that allows users to understand how to use the software and developers to understand the software functionality, business logic, structure, data used, and the technical requirements needed to run and maintain the software. Although the definition does not explicitly describe coding best practices or software development best practices, it mentions documentation, which is a core aspect of best practices.

On the other hand, the definition lacks descriptions of important aspects of what software is. For example, it does not define software as a product, which it is, and what software can do, such as create other products (e.g., a game engine used to create video games) and provide services. Additionally, it does not mention the automated nature of software (capable of doing tasks automatically), which, with the emergence of agentic AI (autonomous software), probably in the near future, would be an integral part of what software is.

It is difficult to define software because it is an adaptive and constantly evolving technology. What started as sets of instructions used to compute calculations (e.g., calculators) have now become incredibly complicated systems, which are deeply integrated into nearly every aspect of modern society. Moreover, with the emergence of agentic AI, they are becoming autonomous systems capable of making decisions and acting on their own, changing the definition of what software is.

The textbook provides the following seven broad categories of computer software:

1. **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. Other systems applications (e.g., operating system components, drivers, networking software, telecommunications processors) process largely indeterminate data.
2. **Application software**. Stand-alone programs that solve a specific business need. Applications in this area process business or technical data in a way that facilitates business operations or management/technical decision making.
3. **Engineering/scientific software**. A broad array of “number-crunching” or data science programs that range from astronomy to volcanology, from automotive stress analysis to orbital dynamics, from computer-aided design to consumer spending habits, and from genetic analysis to meteorology.
4. **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).
5. **Product-line software.** Composed of reusable components and designed to provide specific capabilities for use by many different customers. It may focus on a limited and esoteric marketplace (e.g., inventory control products) or attempt to address the mass consumer market.
6. **Web/mobile applications.** This network-centric software category spans a wide array of applications and encompasses browser-based apps, cloud computing, service-based computing, and software that resides on mobile devices.
7. **Artificial intelligence software**. Makes use of heuristics5 to solve complex problems that are not amenable to regular computation or straightforward analysis. Applications within this area include robotics, decision-making systems, pattern recognition (image and voice), machine learning, theorem proving, and game playing.

(Pressman and Maxim, 2020, p.7)

As explained previously, software is an adaptive and constantly evolving technology, consequently, software engineering is also; so, the core adaptive approach of software engineering can be applied to each category of computer software; however, the way this adaptive approach is applied must be tailored to the constraints, requirements, and stakeholder needs for each project and within each category.

-Alex

References:

Pressman, R. & Maxim, B. (2020). Chapter 1 Software and Software Engineering. *Software engineering: A practitioner's approach* (9th ed.). McGrawHill.