**What is UML**?

UML is an acronym that stands for **Unified Modeling Language**. Simply put, UML is a modern approach to modeling and documenting software. In fact, it’s one of the most popular [business process modeling techniques](https://tallyfy.com/business-process-modeling-techniques).It is based on **diagrammatic representations** of software components. As the old proverb says: “a picture is worth a thousand words”. By using visual representations, we are able to better understand possible flaws or errors in software or business processes.UML was created as a result of the chaos revolving around software development and documentation.

. **What is the use of UML?**

Mainly, UML has been used as a general-purpose modeling language in the field of software engineering. However, it has now found its way into the documentation of several [business processes](https://tallyfy.com/business-process) or [workflows](https://tallyfy.com/what-is-a-workflow/). For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficacy.

UML itself finds different uses in software development and business process documentation:

#### **Sketch**

UML diagrams, in this case, are used to communicate different aspects and characteristics of a system. However, this is only a top-level view of the system and will most probably not include all the necessary details to execute the project until the very end.

* **Forward Design** – The design of the sketch is done before coding the application. This is done to get a better view of the system or workflow that you are trying to create. Many design issues or flaws can be revealed, thus improving the overall project health and well-being.
* **Backward Design** – After writing the code, the UML diagrams are drawn as a form of documentation for the different activities, roles, actors, and workflows.

**Communication UML diagram**

In UML 1.x, communication diagrams used to be called collaborative diagrams. As the name suggests, the main focus of this type of UML diagram is on communication between objects.

Since the core components are the messages that are exchanged between objects, we can build communication diagrams the same way we would make a sequence diagram. The only difference between the two is that objects in communication diagrams are shown with association connections.

Visually, the two differ in that sequence diagrams are well-structured vertically and the message flow follows a top-down chronological approach. Communication UML diagrams on the other hand use number schemes and pointing arrows in order to depict the message flow.

If you would have to choose between the two when writing documentation for a process or system, sequence diagrams would probably be a better choice. Many software engineers prefer sequence diagrams not only because they are better structured, but also because they have been given more attention in terms of the available annotations within the UML documentation.

On the other hand, communication diagrams are much easier to design because you can literally add an object anywhere on the drawing board. After all, in order for objects to be connected, they only need to be part of the numbered sequence, without having to be physically close to each other.

1.1 Temperature ()

1.void loop ()

1.2 pulse rate ()

User 1.3 EEG ()

1.4 Oxygen level ()