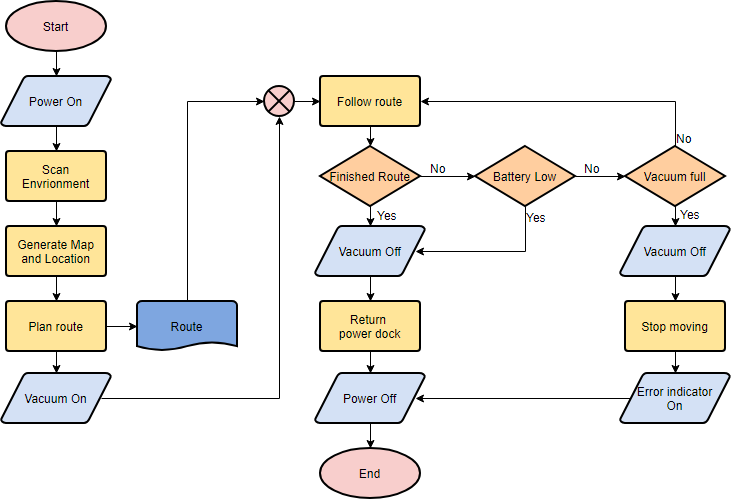
Sanchez, Sophia Anne V.                                                                                                    PROGCON   
TM191/HRO191                                                                                                              October 10, 2019

When do we use them and why is it important?

**FLOWCHARTS**

A flowchart is simply a graphical representation of steps. It shows steps in sequential order and is widely used in presenting the flow of algorithms, workflow or processes. Typically, a flowchart shows the steps as boxes of various kinds, and their order by connecting them with arrows.



**When to Draw Flowchart?**

Using a flowchart has a variety of benefits:

* It helps to clarify complex processes.
* It identifies steps that do not add value to the internal or external customer, including delays; needless storage and transportation; unnecessary work, duplication, and added expense; breakdowns in communication.
* It helps team members gain a shared understanding of the process and use this knowledge to collect data, identify problems, focus discussions, and identify resources.
* It serves as a basis for designing new processes.

**Flowchart Symbols**

Different flowchart shapes have different conventional meanings. The meanings of some of the more common shapes are as follows:

***Terminator***

The terminator symbol represents the starting or ending point of the system.

Flowchart symbol: Terminator

***Process***

A box indicates some particular operation.

Flowchart symbol: Process

***Document***

This represents a printout, such as a document or a report.

Flowchart symbol: Document

***Decision***

A diamond represents a decision or branching point. Lines coming out from the diamond indicates different possible situations, leading to different sub-processes.



***Data***

It represents information entering or leaving the system. An input might be an order from a customer. Output can be a product to be delivered.

Flowchart symbol: Data

***On-Page Reference***

This symbol would contain a letter inside. It indicates that the flow continues on a matching symbol containing the same letter somewhere else on the same page.

Flowchart symbol: On page reference

***Off-Page Reference***This symbol would contain a letter inside. It indicates that the flow continues on a matching symbol containing the same letter somewhere else on a different page.

Flowchart symbol: Off page reference

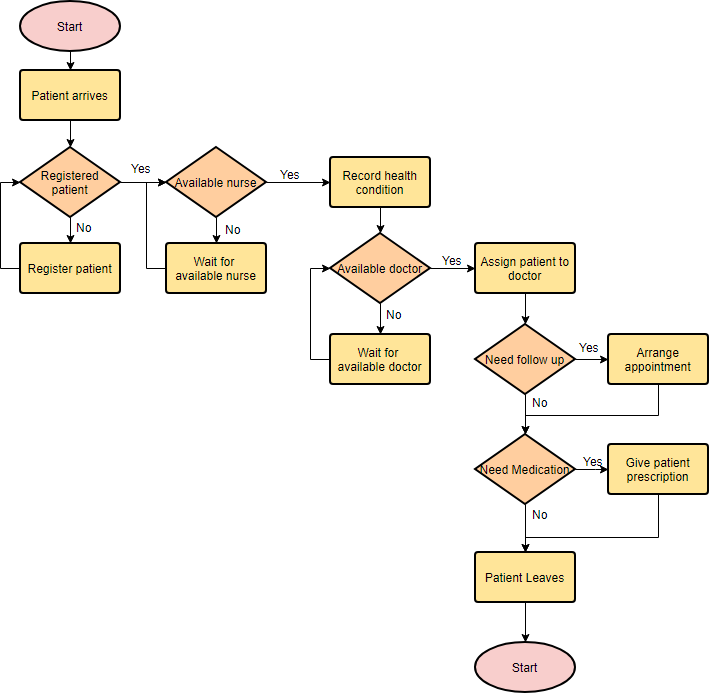
***Delay or Bottleneck***

Identifies a delay or a bottleneck.

Flowchart symbol: Delay

***Flow***Lines represent the flow of the sequence and direction of a process.

Flowchart symbol: Flow

**Flowchart example**

**PSEUDO CODES**

Pseudocode is an informal way of programming description that does not require any strict programming language syntax or underlying technology considerations. It is used for creating an outline or a rough draft of a program. Pseudocode summarizes a program’s flow, but excludes underlying details. System designers write pseudocode to ensure that programmers understand a software project's requirements and align code accordingly.

Pseudocode is not an actual programming language. So it cannot be compiled into an executable program. It uses short terms or simple English language syntaxes to write code for programs before it is actually converted into a specific programming language. This is done to identify top level flow errors, and understand the programming data flows that the final program is going to use. This definitely helps save time during actual programming as conceptual errors have been already corrected. Firstly, program description and functionality is gathered and then pseudocode is used to create statements to achieve the required results for a program. Detailed pseudocode is inspected and verified by the designer’s team or programmers to match design specifications. Catching errors or wrong program flow at the pseudocode stage is beneficial for development as it is less costly than catching them later. Once the pseudocode is accepted by the team, it is rewritten using the vocabulary and syntax of a programming language. The purpose of using pseudocode is an efficient key principle of an algorithm. It is used in planning an algorithm with sketching out the structure of the program before the actual coding takes place.

**Advantages of Pseudocode**

Programming languages are difficult to read for most people, but pseudocode allows nonprogrammers, such as business analysts, to review the steps to confirm the proposed code matches the coding specifications. By writing the code in human language first, the programmer safeguards against leaving out an important step. Some programmers write pseudocode in a separate document, while others write directly in the programming language using comments before the actual code. This provides a handy reference during coding.

**Disadvantages of Pseudocode**

While pseudocode is easy to read, it does not provide as good a map for the programmer as a flowchart does. It does not include the full logic of the proposed code. A flowchart includes decision structures and looping structures in the code, with indications of the next step, depending on the answer to the previous step. Since it is basic by nature, pseudocode sometimes causes nonprogrammers to misunderstand the complexity of a coding project. The lack of standards is probably the main disadvantage of pseudocode. Pseudocode is by nature unstructured, so the reader may not be able to see the logic in a step.

REFERENCES:

Britannica, T. E. of E. (n.d.). Flowchart. Retrieved from <https://www.britannica.com/topic/flowchart>.

Bortolus, D., Southern, M., Dadighat, U., Finch, C., Whatley, T., & Contributor, T. (n.d.). What Are the Advantages & Limitations of Pseudocode? Retrieved from https://www.techwalla.com/articles/what-are-the-advantages-limitations-of-pseudocode#targetText=By writing the code in,a handy reference during coding