Prioritizing

Tom Rochette <tom.rochette@coreteks.org>

July 24, 2025 — b5600af2

1 Steps

- Build a list of task/items
  List everything that you want to get out of your head. The goal here is to make explicit as much as possible.
- Deconstruct tasks into their pre-requisites and follow-up tasks
  There are a couple of important things to consider when one wants to prioritize their task list. One is that even if a task is at the top of the list, it might not be possible to do it until its dependencies are fulfilled. This in turn means that all dependencies will have a superior priority to this task automatically. However, it frequently happens that what we consider dependencies can in fact be delayed or temporarily replaced by another solution which takes less time to implement or costs less (or for whatever other reason can replace the original dependency).
- Split tasks into 2 groups (and repeat this process)
  The idea here is to quickly filter out as many tasks as possible. As you may have noticed, I have not specified the filtering predicate. It is up to you to filter out your tasks such that you will have the least amount to filter at once. Examples of predicates you could use are "will/will not do", "want/do not want", "need/do not need", "like/do not like" and so on.
- Prioritize the tasks that will have to be done
  After a certain number of iterations of the previous step, you should arrive at a point where the items you have all need to be done, but you do not know in which order you have to do them (or want to do them).

2 Methods

2.1 4 quadrants method

Urgent/Important
Not Urgent/Important
Urgent/Not Important
Not Urgent/Not Important

2.2 Analytic Hierarchy Process

2.3 Binary Search Tree

2.4 Planning Game

2.5 100-Point Method

3 Prioritizing reading

- What you feel like reading

- Reading dependencies
- ROI evaluation

4 References

- Karlsson, Joachim, Claes Wohlin, and Björn Regnell. "An evaluation of methods for prioritizing software requirements." Information and Software Technology 39.14 (1998): 939-947.
- Karlsson, Joachim, Stefan Olsson, and Kevin Ryan. "Improved practical support for large-scale requirements prioritising." Requirements Engineering 2.1 (1997): 51-60.
- Ahl, Viggo. "An experimental comparison of five prioritization methods: investigating ease of use, accuracy and scalability." (2005).
- Gill, Nasib Singh. "A Comparison among Various Techniques to Prioritize the Requirements." International Journal of Computer Science and Management Studies (IJCSMS) www. ijcsms. com 1.12: 601-607.
- http://www.gwern.net/Resorter