CRITICAL THINKING TASK

The tradeoff between time and resources in computer science refers to the balancing act that software developers must undertake when deciding how to allocate limited resources, such as computational power, memory, and development time, to achieve desired outcomes.

In this tradeoff, "time" represents the speed and efficiency of completing tasks, including development time, execution time, and time-to-market for software products. On the other hand, "resources" encompass the various tangible and intangible assets required for software development and operation, such as CPU cycles, memory usage, financial resources, and human capital.

Software developers face this tradeoff when choosing between different approaches, technologies, or strategies for developing software systems. For example, opting for a high-level programming language may accelerate development time but could result in higher resource consumption, while using a low-level language might require more development time upfront but lead to more efficient resource utilization in the long run.

<u>3 Different Metrics to Compare:</u>

1) High Level Language – Uses a lot of Computer Resources (CPU and Memory), Almost no Company resources used, does not require lots of time to code, Is the most reliable method of developing

2) Low Level Language – Uses less Computer Resources (CPU and Memory), Almost no Company resources used, require lots of time to code, Is not the most reliable method of developing

3) Development Using AI – Uses least amount of Computer Resources (CPU and Memory), Heavy Company resources are used, takes the least amount to code, Is least reliable method of developing