3. Theories

Question 1: With your own words, please describe shortly what is Network Effects Theory (according to the course book)? In addition, based on your experiences and previous learning, have you noted the direct or indirect network effects working previously in the software industry?

Network Effects Theory is the idea that the value of a product or service depends on the number of users who use it or are compatible with it. The more users there are, the more valuable the product or service becomes to each user. This creates a positive feedback loop that attracts more users and increases the value further. There are two types of network effects: direct and indirect. Direct network effects occur when the value of a product or service increases with the number of users of the same product or service. Indirect network effects occur when the value of a product or service increases with the number of users of a different but related product or service.

Microsoft Windows is a typical example of a software product that exhibits direct network effects. It has gained a large market share due to its compatibility with various hardware and software products, and its ability to run a wide range of applications.

Question 2: With your own words, please describe shortly what is Transaction Cost Theory (according to the course book)? In addition, based on your experiences and previous learning, do you see Transaction Cost Theory relevant to the software industry?

Transaction Cost Theory is the idea that there are costs associated with making any economic exchange, whether it is with an external party or within an organization. These costs include the time and effort of searching, negotiating, monitoring, and enforcing the terms of the exchange. Transaction Cost Theory suggests that the optimal way of organizing an exchange depends on the characteristics of the transaction, such as its frequency, uncertainty, and asset specificity.

I believe that the Transaction Cost Theory is still relevant to the software industry. It can help explain the choices and trade-offs that software companies face when deciding whether to produce, buy, or collaborate on software products. For example, a software firm may choose to develop its own software internally, if the software is highly specific to its needs and requires frequent and uncertain interactions with the developers. Alternatively, a software firm may choose to outsource or license its software from another firm, if the software is standardized and does not require much customization or cooperation. A software firm may also choose to form a strategic alliance or a joint venture with another firm, if the software is complex and involves proprietary assets and capabilities.

Question 3: With your own words, please describe shortly what is Principal-Agent Problem (according to the course book)? In addition, based on your experiences and previous learning, do you agree with the premises of this problem? Why? Do you have some experience to share which (could) have been solved by acknowledging the PA Problem?

The Principal-Agent Problem is the idea that there is a conflict of interest between a person or group (the principal) and the person or entity that acts on their behalf (the agent). The agent may have different goals, preferences, or information than the principal, and may not act in the best interest of the principal. This can result in agency costs, which are the losses or inefficiencies caused by the agent’s actions.

I agree with the premises of this problem, because it is a common and realistic situation in many contexts, such as business, politics, law, and education. The problem arises when there is a separation of ownership and control, and when there is asymmetric information or moral hazard between the principal and the agent. One example of a Principal-Agent Problem that I have encountered is when I hired a contractor to renovate my garden. The contractor (the agent) had more knowledge and expertise than me (the principal) about the quality and cost of the materials and labor. The contractor also had an incentive to maximize his profit by using cheaper materials, cutting corners, or delaying the project. I had to monitor the contractor's work and performance, and use a contract that specified the terms and conditions of the project, such as the budget, timeline, and quality standards. This way, I tried to align the contractor's interests with mine, and reduce the agency costs by using a performance-based contract. This means that the contractor is paid based on the quality and timeliness of the work, rather than on the hours or materials used. This way, the contractor has an incentive to deliver the best possible outcome for me, and I can avoid paying for unnecessary or substandard work.