Ulat, Roma Alyanna M.

Activity #2

Using the law of Peter Senge's Laws of Systems Thinking, give an explanation and example in problem-solving skills in the context of software development.

1. “Today’s problems come from yesterday’s solutions

Doing “quick fix” to a certain situation can sometimes lead to bigger problems. Decisions we make today can result to future’s struggle. This can cause stress later on either for the people who made the decision or to other parts of the organization. For instance, you were tasked by your boss to develop a software for a client. Since you want this software to be developed quickly and to keep cash coming in the short term, it was poorly scoped and poorly defined at low budget. Months after, your team is still stuck on working on that project and it already costs too much money and time with no chance of profit while the client is unsatisfied. To avoid this problem, we should take “one step at a time.” We should not rush things and think carefully. Use your resources wisely. If you think that your system or project will need of high budget to achieve a more desired result, go for it. Because if you save too much, it can cause you more money to use in the later part.

2. “The harder you push, the harder the system pushes you back.”

This law reminds us that sometimes, our best intentions and efforts to grow and push harder to achieve something we desire does not create the result we want. The harder you push, the worse it seems to become. Most of us push way harder through risky and tricky situations, especially when things are not leading to what we had hoped for. For instance, you push harder in developing poorly-priced projects or software in order to keep cash coming in a short span of time. The said projects need many revisions. Because of this, more people are needed to get the job well done. By hiring more people, the higher the expenses. Instead of looking into the bigger picture, you want to achieve something in a short term. To prevent having issues, make a better plan. Again, don’t save too much money if this will only lead you to something worse.

3. “Behavior grows better before it grows worse.”

Short term solutions give temporary improvement but never remove or eliminate the root cause of the issue. This will make the situation worse in the long run. For instance, you are tasked to develop a system for a client. Since you are struggling with money, you use cheaper raw materials to develop this system. You successfully finished the project but in the long run, this project needs many changes because it was poorly designed which will cost you more. To prevent this, don't settle for less. You need to make a plan. Resist the temptation to apply any solution until you thoroughly understand the root causes of the problem.

4. "The easy way out usually leads back in."

The best solution often requires systematic or fundamentally different approaches to eliminate the problem. It is important to understand more about the problem and how eliminating it will affect other related systems to avoid creating bigger problems. The example for this can be the situation in number 3 wherein you eliminate the problem by using cheaper raw materials which in turn leads to a bigger problem. You feel like this is the quicker and obvious solution without considering the bigger picture. In this case, the solution should consider the consequences before adopting a solution.

5. "The cure can be worse than the disease."

The solution can be worse than the original problem if the main issue is not well-defined. Sometimes, your proposed solution is not only ineffective but also dangerous. You need to think if your solution will fit the existing system. For instance you created a voice assistant app, with the power of AI, that will help visually impaired and blind people to see things clearly and live independently. Instead of helping users, it gives them more problem because the voice on the app is describing the surroundings wrong which leads users to accidents. To solve this problem, the developer needs to test the application carefully or transform it to live assistance app wherein the users will receive live assistance from sighted volunteers.

6. "Faster is slower."

It is tempting to advance at full speed without caution at the first taste of success. Remember, if the solution aims to increase the system productivity beyond its optimal rate, the system may actually slow down to compensate for this change in growth rate. If the problem increases the rate

over the optimal growth rate, then the system will slow down its growth to reach the optimal growth rate. For instance, your client is already upset about the software she's been waiting for a month already. You pushed yourself and your team to work harder for a longer period of time. With this, the system is successfully finished but you end up with a burnt out team who slow down their progress in order to overcome their exhaustion. To prevent this, don't force yourself and take one step at a time. If you think you need more people in your team to finish the project, go for it.

7. "Cause and effect are not closely related in time and space."

When trying to solve a problem, look for underlying, systematic issues not just for immediate causes. The cause or number of causes may create effects that are at different times and locations. For instance, you were able to finish the system for your client but it was poorly designed and developed. Because of this, you blame the TIME because you only have a short period of time to finish the system when in fact there were several decisions that all factored into the problem like tight budget, lack of people, poor planning etc. Find the root causes of all the problem. Don't settle for one. The most important thing for you is to look at the bigger picture.

8. "Small changes can produce big results - but the areas of highest leverage are often the least obvious."

Small changes can lead to bigger result. In other words, small, well-planned solutions can offer significant impacts, especially when they are implemented carefully in a right time and place. For example, finding the right combinations of procedures to solve a problem that can contribute to building a long-term, sustainable solution.

9. "You can have your cake and eat it too - but not all at once."

Solvers based their decisions on the "either or" selection method. In many instances, we think something is an either/or problem when in fact its a dilemma that can become both/and if we change how we think of the problem and allow time for solutions to work. In this context, we need to understand the problem, the company, and if the problem is actually an effect of a bigger issue.

10. "Dividing an elephant in half does not produce two small elephants"

Problems need to be seen as a whole not as individual parts. Separating problems into parts can lead to more problems. Solution should be planned with consideration for the whole organization. For instance, your team was tasked to develop a software for a client. The team members include - the designers, developers, producers, and whoever plays an important role for this project. All members have different suggestions about the software which leads to argument. To prevent this, the team leader should consider all the suggestions.

11. “There is no blame.”

Usually, once the problem gets worse, people tend to point a finger at someone as the sole guilty person. In systems thinking, everyone is part of a whole system. The solution should consider the community while planning the solution. For example, you need to develop a software or app that will help the community. In order to learn about each one’s perception and suggested solutions, it requires involvement of the community.