Systems thinking is all about understanding the relationships between things, people, and events. If we think about our master’s program, cloud-based software engineering, as a system, we must first identify some of its parts.

Some things that make up the program include the curriculum with all its courses and the final thesis, resources given by the teachers, and assignments that we must hand in. Some events that occur during the program include lessons every week, final exam at the end of the course, and ever occurring learning that it’s all about. Learnedness can also be seen as a thing that grows.

A behavior over time diagram can be used to represent learnedness. As we go through courses our skills and knowledge accumulate. Working on the thesis adds to this in a great way, increasing the rate of learning.   
Kuva, joka sisältää kohteen diagrammi, viiva, Tontti, Fontti

Kuvaus luotu automaattisesti

Another important factor that is present in the master’s program, are people. Some of the key people are the teachers are students. They play a vital role in the system as they do not only interact with each other; they also interact with every other event and thing in the system.

Being in the role of a student can be stressful. Learning is hard work as lessons and assignments take a lot of time and effort. From the very beginning, the importance of a support network has been emphasized. Different tasks add to stress level and sometimes a little help from a friendly face can go a long way in helping students along. This interaction can be described by a causal loop.

Kuva, joka sisältää kohteen diagrammi, teksti, ympyrä, luonnos

Kuvaus luotu automaattisesti

In the diagram, we can see a balancing loop of stress level being kept in check by performing tasks given by teachers. If the number of assignments increases rapidly in relation to a student’s capacity to perform them, stress levels can begin to approach unhealthy levels. Asking help from peers, or the teacher can yield helpful tips that help perform tasks more effectively.

In the diagram we can see a double o. This means that a vicious or a virtuous cycle can form. Always asking for help instead of struggling to produce results on your own can cause learning debt. This debt can make following assignments more difficult. On the other hand, a clarifying question can unlock a knot in your brain and help in unexpected ways.

If we look at courses, they can seem like a collection of skills. If you zoom out a little bit and think of working life, it becomes clear that they are skills needed by a master level engineer. The courses touch on technical subjects and leadership. A leader in the software field must be technically knowledgeable as well as a good leader, who understands people and different styles of management. In a system, purpose plays a role. The collection is intentionally made to cover skills that benefit master level software engineers. This makes the collection a system.

Other things that define a system include that the parts present cannot be changed, and the order of the parts change the outcome of the system. The presence of one optional course makes the status of the system slightly questionable. I would argue however that the system instead just becomes a slightly different system. The order has been carefully selected to ensure the best learning outcomes. If it’s true or not, is not important, the important thing is that someone thought that was the case.

Another important feature of a system is that it attempts to maintain stability through feedback. This one is difficult to justify if looking strictly at the contents of the curriculum. It does, however, fit in if you think of the courses as something that must be completed, and each course is a system in its own right. Each course has contents that vary depending on how the students perform. If performance is low, it’s better to not push them over the edge and have them learn the most important parts well. This variance caused by feedback propagates upward into the larger system.