# Part 1

At the first week of the “Software Engineering: Introduction” course I have been talked about processes, their pros and cons, and about the difference between them. So, as I learned, there are two major categories of processes: Traditional and Agile. In our days any traditional approach is rarely used because of it’s development velocity. In the best scenarios user’s feedback will be given only after a year from the start of your development iteration. But it could be given right away. Despite this valuable shortcoming, Traditional approach can be used in a development of a small system. In this case Traditional approach suits more than Agile one because of complexity of any Agile methodology.

In my opinion the most useful approach for me is Scrum. I see myself as a programmer in a company that develops complex solutions, so that means no Traditional approach is acceptable. The other thing I’ve chosen Scrum is because of it’s popularity and convenience. You will always have shorter deadline and easier tasks to be done, so that will increase your productivity.

To be honest, I already knew all this material before, but still had some problems with understanding the material that was given. For example, there was a video about the Spiral approach and the lecturer spoke about shortcomings of this approach. He mentioned that Risk Analysis might be quite overwhelming, but he didn’t explain why. Defects such this one, are not so valuable because the rest of the material was clear and easy to understand. Having them fixed will be great though.

As I mentioned before, most of the material was clear and easy to understand. It contains info about not so useful approaches, but without it the picture of the processes wouldn’t be complete. The only thing I’d be glad to see is opened quiz section, which is closed till you have no certificate.

# Part 2

Today I am supposed to talk about Nikita’s report. Nikita’s report was easy to understand, he spoke clearly. As I see it, there is nothing to improve with his language. Despite me being not interested in his aspect of programming, I’d be glad to hear what he learned about.

Yes, he did mention what he learned, but he didn’t mention what it is all about. For example, he talked about some Python library, talked that it works with arrays, but he didn’t talk about why should I use it, what exactly I can do with it.

What I am trying to say is that he only talked about structure of his course, but not about it’s components particularly.