

## Why was Lyee created?

While taking an initiative of 80 projects of software, Mr. Negoro studied about 3,000 papers and thought about what is the best methodology for the software development.

In a conventional methodology, people believe that software programs are easily produced if requirements are clearly defined.

However, there are fundamental problems in this approach.

- (a) Requirements are written in a natural language, and the meaning of the language used here is ambiguous because each individual has its own interpretation on the requirements.
- (b) And we try to define requirements with empirical knowledge, and to produce programs based on the specifications which are defined in an inevitably ambiguous manner. This is indeed unjustifiable.
- (c) Therefore, each program has a different structure. Relations between requirements and their programs should be uniquely determined in their ideal form, but in actuality they are ambiguous as you can see so that programs become “a black box.” Most programmers are not sure of the contents of their own programs.
- (d) In addition, the requirements themselves often change. This makes software development processes even more complicated to cause low productivity and poor maintainability of programs.

On the contrary, Lyee takes a totally opposite approach to the conventional methods. Lyee determines both requirements and programs automatically by using its universally valid program structure.

In order to explain the details of it, let me begin to look back at the recent trend of software engineering. Since around twenty years ago, researchers have realized that all the software problems could be resolved if they were to find out the smallest unit which could be commonly used for all programs and make programs with that unit. DOA engineers, in particular, tried to make programs for each word of requirements. The word used here means data item for input and output. However, all the attempts resulted in failure.

Mr. Negoro was one of them to pursue the same goal, by conducting researches and trying to find that smallest unit. In the first place, he

divided the conventional development process into approximately fifty steps. He could in effect double the productivity in this way, but he failed to figure out what that magic unit was.

He then thought that if he were able to divide the process into, say, one hundred and fifty steps, he was able to specify the unit. But it turned out to be impossible to do so. One hundred and fifty steps were too many. Through these trials and errors, however, he realized that the unit was a word because the software was a set of data or words. To set a value in data area could be said to form a meaning. Suddenly he got an inspiration that software is meaning in its ultimate sense.

In order to determine the universally valid program structure for every word, he thought he must take a semantic approach rather than software engineering one, and he finally succeeded in establishing Lyee methodology and theory.

“Meaning” is intrinsically arbitrary and constantly changing. All the problems related to software are always caused by its arbitrariness. But, he thought that this precarious characteristic was appearing only in the natural space. That is, there must exist what he calls a consciousness space, which is an original space to create static meaning with outlines, but we can never recognize it. If we could trace back to that space, we may be freed from the arbitrariness of the meaning.

Mr. Negoro first set logical atoms as the smallest but ultimate existence to form meaning. By the given providence, logical atoms are gathering to make sets to create the consciousness space and the cognition space. The cognition space is consisted of four sub-spaces. He called them initial, actual, power, and hyper-power sub-spaces in the order of the formation. But, except for the hyper-power sub-space or natural space in another name, we cannot recognize these sub-spaces.

In sum, to find out its meaning in the consciousness space through a word in the natural space is to put the consciousness space and the natural space in a critical state. These two spaces are bridged. If applying these bridged spaces to software structure, we can gain a universally valid program. This is when Lyee is born.

It may look nonsensical to have hypothesized these above-mentioned spaces except for the natural space. If you look at revolutionary effect produced by the Lyee’s methodology, however, it is highly possible to prove that this assumption is true.