Kai-Fu Lee is a famous computer scientist from china. He has participated in and made great contributions in many remarkable software projects during his career in several companies such as Apple, Microsoft, Google.

He developed the world’s first speak-independent, continuous, speech recognition system as his PhD work at Carnegie Mellon University. In his “Large-vocabulary speaker-independent continuous speech recognition: The SPHINX system” article which was published in 1988, he pointed out that most of the speak recognition systems merged in the past decade all have some constraints to ensure the accuracy. From his point of view, the ultimate speech recognition system should not be bounded by the constraints then as his article title mentioned: 1. large-vocabulary, 2. speaker-independent, 3. continuous would be his “SPHINX” speech recognition system’s major breakthrough in this specific area.

As far as I am concerned, the development of technology or certain software could be step by step or gradual. During the development process, constraints are inevitable and sometimes they are the key to make evolutions. People may develop some prototypes at the very first stage which potentially ignore the constraints or simplify the real situation to make the software work, then they may try to modify or recode the program to satisfy various constraints later. This kind of development is kind of similar to the agile development while the constraints or requirements are proposed by the customer during the development and developers consecutively fix or add new functionality to the old version. The three major breakthroughs mentioned above is to make the software or the system more general and fit different environments. According to Lee’s own test, his SPHINX system gained 96% accuracy on a 997-world task which is a remarkable achievement at that time and was awarded the “Most important Innovation of 1988” by Business Week magazine.

Lee joined the Silicon Graphics (SGI) in 1996 and participated into the OpenGL project. OpenGL is a software API to render 2D and 3D vector graphics, it can interact with the GPU to boost the rendering and it’s a cross-language and cross-platform application. At that time, Lee tried to implement 3D technology on normal PC and don’t depend on the advance hardware configuration. Like the project in his PhD, his purpose is to make the technic more general because the machine made by SGI before is expensive so it’s hard to implement 3D on normal PC and the product’s customer is limited to some special government departments. His idea is to render 3D graphic in browsers on PC without high hardware configuration. When he displayed this OpenGL product to the public, people were shocked by his creativity. Later he added more functionality to enhance the application and extended the support to more platforms but the corporation with Microsoft has been failed since Microsoft already has its own rendering application. In his own company SGI, the new boss also had doubt on this product’s true potential business value. Although OpenGL is much better than Microsoft’s own application which can only be run on Windows operating system, Microsoft don’t want their technology to be restricted by other company.

Lee learned from this failure that the value of certain products can’t always be money, in addition, the estimation can’t depend too much from a developer’s view. According to his own conclusion, the point is that when we are trying to make some advance products, sometimes we just want to make the product better and better and but forget that the product has to be published to the public who don’t have many specific knowledge in the end. The true value of the product can only be known to the developer, the public might not pay too much attention on it, what they do care might be the practicability or convenience only. From his experience, what inspires me most is that when we are trying to make projects and doing programming, we can’t isolate us with the public, this sometimes can be done with designing user-friendly interface or writing specific manual to introduce the product in a simple way. Also, suppose we are doing agile development, we should have the ability to translate customers’ simple requirements to complex implementation, but then we have to use simple words to explain the complex implementation to the customers and verify how the products satisfy the requirements. In short, a software engineer should not only know how to produce excellent codes, but also how to communicate with customer and introduce the product to the public using simple words.

Lee also founded a website for Chinese teenager to learn the internet in 2004 and wrote several public letters to Chinese college students to share his own experience. After he left google, he founded a company to help people start business in china. Nowadays, he is trying to make contribution on AI field and help the humanity to embrace the benefits of AI. To conclude, Lee is a man who not only good at his own field, but also concerns the whole society’s charity. We might not be able to be a man like him in the future, but at least we can learn from his spirit and try to do our own contribution to the society.