XP

On Monday, September the 15th, Dr. Robert, as a guest speaker, introduced to us an effective way to develop software using software engineering. The speech was about XP, extreme programming. He is currently working with the development team of Profit Star's company. The development team was having difficulties in completing their tasks. In order to overcome the existing problems, Dr. Robert, using XP, suggested changing the methodology of the team. His suggestions mainly targeted on the team's communication among members and the team with the customers. Pair programming was also introduced to improve productivity. These two aspects have drawn my attention that they would be practical and likely to be accepted.

The greatest improvement was the increased sufficiency in communication among development team. Before the team started XP, programmers and testers are located in different rooms, and they rarely talked to each other in person. This resulted in an unfriendly environment and difficulty building teamwork. So he suggested the team being located all in one room, and they should be able to see each others. To increase communication among programmers, the coders were using the bullpen, where four of them per round table. This let the ideas flow more quickly because the programmer could ask questions and receive immediate feedback without having to leave the spot. This sophisticated method is practical in details and easily adapted.

Another technique to improve team’s communication is pair programming. The programmers were divided into group of two to work on the project. This increases the relationship between the programmers and also promotes knowledge diffusion. Eventually there will not be a person who is the only one knows how to do something. The method apparently raised the average understanding of the team on the running system. This may seem slower than each person working individually on different projects. This is true for the typing process; however, we all know that it does not take the noticeable time.

The speech has broadened the range of possible outcome we can obtain with good practice of software engineering. The team has good programmers; however, they could not gain success without proper approaches to the problems and better methodology. In conclusion, software engineering has slowly, but steadily, become a significant part of computer science, or technology as a whole.