

Software Engineering homework#1

17011599 안정연

1.To reduce costs and the environmental impact of commuting, your company decides to close several offices and to provide support for staff to work from home. However, the senior management who introduce the policy are unaware that software is developed using agile methods, which rely on close team working and pair programming. Discuss the difficulties that this new policy might cause and how you might get around these problems.

If there is a problem while developing software, team members have to solve the problem by themselves due to difficulties in communicating. Therefore, solving the problem takes a long time and this leads to a decline in productivity. Also, things don't go on because we don't know how much each worked and how much they developed. The key to agile methods is cooperation and feedback. Telecommuting is bound to take longer than meeting. The changes can lead to slow the overall development of the project.

To solve these problems, they need to have a videoconferencing or use chat tools to communicate right away. It also requires developers to use software that can easily share and manage source code.

2.Historically, the introduction of technology has caused profound changes in the labor market and, temporarily at least, displaced people from jobs. Discuss whether the introduction of extensive process automation is likely to have the same consequences for software engineers. If you do n't think it will, explain why not. If you think that it will reduce job opportunities, is it ethical for the engineers affected to passively or actively resist the introduction of this technology?

I think the introduction of extensive process automation is likely to have the same results for software engineers. But I agree that it is not ethical for the affected engineer to refuse to introduce the technology. Software engineers are responsible for engineering jobs and society. They shouldn't simply be about technical matters. The automation system has changed our society a lot. It has not only increased speed but also increased quality. If software development is also possible, it will be able to develop faster and more accurate than engineers, and software users will be more comfortable. When software engineers reduce mechanical tasks, they will become more innovative and creative in their tasks. I think software engineers should fulfill their responsibilities and obligations to society with more innovative developments.

3. Giving reasons for your answer based on the type of system being developed, suggest the most appropriate generic software process model that might be used as a basis for managing the development of the following systems:

a. A system to control anti-lock braking in a car.

It seems the waterfall model is the most appropriate generic software process. The waterfall process is usually adapted for safety-critical systems because of the higher amount of analysis and documentation required before implementation. This is a primary safety-critical system, which has static requirements, and which is tied to a hardware delivery date, so a waterfall process is most appropriate.

b. A virtual reality system to support software maintenance.

The most appropriate software process for a virtual reality system to support software maintenance is Incremental development. The requirements of system will change and cannot be predicted before the implementation. Software maintenance is best done through versions and updates over long amounts of time, which would be the incremental development process.

c. A university accounting system that replaces an existing system.

This is mission-critical system and the requirements are static and reusable (The requirements of system can be predicted in advance because of the existing system). Therefore, Waterfall model is the most appropriate.

d. An interactive travel planning system that helps users plan journeys with the lowest environmental impact

I think this could be more than one of the software process models, so the prototype model and incremental development is the most appropriate. The user's requirement may likely to change and fast delivery is essential to be implemented. A planning system would need to be adaptable and have multiple versions.