CPT203 Software Engineering 1

Tutorial: Agile Methods

Suggested answers

1. In Rapid Software Development, one of the fundamental characteristics is Minimum Documentation. Explain why minimum documentation is important in this context? Which aspect of the software development must be documented and why?

   Answer

   In a Rapid Software Development, engineers argue that documentation is a waste of resources because if the requirement is changed, all effort put on developing documentation will be wasted. If we produce a simple, logical, and concise design and software code, it will naturally act as the documentation.

   Documentation for system requirements is the utmost important documentation for a rapid software development. It acted as the contract between and users and the development team. It acted as the direction and objective for the development.

2. Explain how the principles underlying agile methods lead to the accelerated development and deployment of software

   Answer

   The principles underlying agile development are:
   - Individual and interactions over processes and tools. Individuals interact face-to-face is always more efficient that having to use process such as formal meeting, reading memorandum, and so forth. This means that the team with high efficiency in communication can focus more on producing useful working software.
   - Working software over comprehensive documentation. This contributes to accelerated development because time is not spent developing, checking and managing documentation. Rather, the programmer’s time is focused on the development and testing of code.
   - Customer collaboration over contract negotiation. Rather than spending time developing, analyzing and negotiating requirements to be included in a system contract, agile developers argue that it is more effective to get feedback from customer’s directly during the development about what is required. This allows useful functionality to be developed and delivered earlier than would be possible if contracts were required.
   - Responding to change over following a plan. Agile developers argue (rightly) that being responsive to change is more effective than following a plan-based process because change is inevitable whatever process is used. There is significant overhead in changing plans to accommodate change and the inflexibility of a plan means that work may be done that is later discarded.

3. When would you recommend against the use of an agile method for developing a software system?

Answer

Agile methods should probably not be used when the software is being developed by teams who are not co-located. If any of the individual teams use agile methods, it is very difficult to coordinate their work with other teams. Furthermore, the informal communication which is an essential part of agile methods is practically impossible to maintain unless they are being supported by collaboration tools such as online meeting.

Agile methods should probably also be avoided for critical systems where the consequences of a specification error are serious. In those circumstances, a system specification that is available before development starts makes a detailed specification analysis possible.

Other scenario when Agile methods should be avoided are: -

a. Project with well-defined requirement and required high level of controls.
b. Project that required formal process and/or formal documentation.

4. What are the challenges of an Agile development?

Answer

Check the PPT.

5. A customer who is not ready to devote his time and effort to support the Agile team should not take off a software project using Agile methodologies. Explain.

Answer

Agile is an iterative and incremental development. The implication is that the development team will conduct requirements specification iterative throughout the entire project life cycle, that means the customer must repeatedly be working with the development team to specify and fine tune the requirements. Customer who is not able to devote their time and effort throughout the entire project will fail the project.

6. One of the questions an engineer must ask when deciding the balance between a plan-based and an agile approach is “How large is the system?”. What is the significance of asking the question?

Answer

If the system is very large, it must be plan-based or plan-based with agile approach for the sub-systems. Very large system with no plan is as if it has no aims and objectives, and therefore is doomed to fail.