Lab 1 essay

Agile software development was a term first coined in 2001 in the Agile Manifesto. It is a more progressive alternative to the standard process of designing software in response to the rapid evolution of technology. In general, software planning resides on a scale that ranges from adaptive to predictive. Agile development exists on the adaptive side, and focuses on responding to problems and difficulties as they occur, quickly and efficiently. Rather than attempting to avoid issues at every step like the waterfall method, agile development operates on the assumption that problems will arise, and that the software will be fixed and given value based on its nature of overcoming errors. This also reduces the work that must be applied during the maintenance stage, which is the most expensive stage of a software's existence.

Another core difference is that in standard development planning, the testing and building phases are distinctly separate, and there is heavy focus and funding on the testing phase. In agile development, the two phases are integrated. This allows the team of developers to be reactive to the changing value of the program as it evolves. With testing done at every iteration, the team can determine the value of the software at every stage, and the development philosophy becomes more focused on product than project. This is largely beneficial because it allows the team to assess the future capabilities that the software could have. It also, in its wisdom, saves time and money that the standard waterfall method uses going between building and testing phases. The problem is that the scrupulous testing and dedicated revisions offered by the waterfall method can usually produce a higher quality software with solid design.

Critics of agile development attack its lack of focus on comprehensive documentation. They believe it undermines the validity of software that is more carefully designed. Proponents of agile development disagree, stating that too much documentation is often a bad thing because it is too rigid to be efficient, requires too much time, and is often wrong. If a program is very complex, then highly detailed documentation is likely desynchronized with the current state of the program. Agile development uses the philosophy that documentation should be Just Barely Good Enough (JBGE), such that the editor of the software can understand what is going on without being misled by information that is either too dense or wrong altogether.

The waterfall method of designing a program is rather outdated. Programmers today do not use punch cards or more primitive techniques where editing, revising, and debugging was difficult and timeconsuming. Agile development yields a product that is solid, polished, and less likely to contain bugs because of the nature of the development itself. There isn't a reason to use waterfall over agile, and so agile is the better method for designers in the present.