Gerard Andrews

1/24/2017

CMPT 220

Dr. Rivas

Agile Development vs. Software Development Process

Agile Development is a series of principles in Software Development that attempt to fix the problems of traditional software development. Some of its core fundamentals include: Necessary collaboration, flexible planning, fast maintenance, and faster publishing of software. Another core idea is that Agile Development attempts to reduce the risk of projects failing, and can also produce more immediate results.

Compared to the standard Software Development Process, Agile Development is faster when outputting a solution to development. It tries to get something out as soon as possible to help understand what the problem is. This is done with very limited planning. Early release can also stop projects from failing and allow them to be maintained and grow faster. Collaboration and communication also apply to making solutions faster as well. Most importantly it's idea to make a flexible plan for a large portion of time can allow solutions to be made faster. This can also allow the projects to be adapted to a later use overtime.

Agile Development does of course have its shortcomings. An example is the flaw that can be seen in outputting a solution early. By trying to put out a solution early, it may involve not putting any emphasis on designing and documentation. These steps are crucial to understanding the problems and how to effectively make a solution that solves them. The other point is that Agile Development requires a large amount of teamwork, that may or may not be there. If one group isn't communicating, it adds to risk and delays outputting a solution early, which can short circuit the entire point of Agile Development. Customers can also be misleading in what they say which could also put the whole development process off track.

The regular Software Development Process has its merits when compared to Agile Development. First off, it is simple and easy to use because the process is easily understood. Phases are done one stage at a time so some points don't overlap. Finally, The fact that it goes through a process for designing around a problem means that the standard process can expect a better first result than Agile Development. This model is best with small-scale projects

Of course, the process of Software Development has its issues as well. It's a much slower process compared to Agile Development, meaning the solution may not always be there immediately. Another issue is that it's a lot riskier than Agile Development because of the time it takes to create a solution. The planning is potentially not as flexible for future issues because the process is meant to focus on that current issue, adding more to risk. Once it is in testing stage, there is no going back to fixing the problem. This model is not so good for dealing with long-term projects

For me, I prefer the traditional Software Development paradigm, because I prefer a very good first result when designing a solution to a problem. Although maintenance may be more of an issue and it may be riskier compared to Agile Development, I'm willing to take on those risks because going through a process that involves close listening to what people want may allow me to make a more flexible plan. Also I just prefer easy to understand processes, because it eases what could be a time consuming process.