Brandon Kline

Prof. Rivas

CMPT 220L

31 Jan 2017

The traditional software development process follows a structured design philosophy, in which a set number of steps occur in a specific order while the software is being developed. Firstly, the requirement specification phase defines what problem the creation of the software will solve, as well as what exactly the software will do. Secondly, the system analysis phase calculates the data flow and identifies the inputs and outputs of the system. Thirdly, the system design phase creates a process for obtaining the input from the output, as well as breaking down and designing each component of the system. The implementation phase then translates the previously created system design into individual programs, then the testing phase locates and fixes bugs in the code from the implementation phase. Deployment then makes the product available for widespread use, and maintenance is tasked with managing and updating the product.

By comparison, an agile development process involves multiple parts of the development process occurring at the same time and collaboration between different stages of development. For example, requirements specification will occur alongside system analysis, system design will work alongside implementation, and testing and deployment occur simultaneously. Although the steps are almost identical in both development processes, it is the collaboration that occurs for more rapid completion and fluidity in agile development that is the major difference between the two. As a developer, the traditional software development process is better suited for intensive programming work, as though it may be faster to work alongside system design in an agile development process, more time is allotted within a traditional development method and issues between developers and other areas of the project are less likely to occur.