From an academic perspective, the waterfall model probably is not the first choice, as it is incredibly inflexible and requires all the requirements to be known ahead of time. However, our company specializes in software for heavy machinery. We build the software that runs assembly lines and related equipment in factories. The company that builds the machinery from parts is not going to build a new machine if the factory managers change their minds. As such, it is safe for us to assume that the requirements of the software will not change. Furthermore, manufacturing equipment is heavily regulated, due to safely concerns for the works and quality concerns for the products. Therefore, we need a lot of documentation, a solid design, and approval from a government organization to move forward. It would be extremely costly to develop a software product that the overseeing organization refused to approve. The agile model would be more useful if there was a higher probability of evolving requirements. In this case, the agile model would not supply us with sufficient documentation to prove our adherence to safety regulations. I would encourage you not to try to introduce any agile practices, as they simply do not fit the types of products developed in this company. As for adapting to the waterfall model, I would encourage you to look through the requirements when we receive them, think about possible software architectures, and plan out the development of the software. Show your outline to me or one of our team members, and we will help you improve it. Most importantly, remember that you have all the requirements and you are in control of the software. You can do whatever planning is necessary, and you can develop whatever architecture is most sensible for the end product.