

When a client purchases software as a service, they are purchasing the right to use that software. The software developers will provide them with bug fixes and major updates at no additional charge. Often, the software is run by the service provider, significantly reducing the operating costs for the user. Since software as a service is a subscription-based model, it encourages the software developers to try and keep their current clients happily enjoying their software. When a client purchases software as a product, they are purchasing the rights to use a copy of the product. In order to update the software, the client would have to pay for the new version, instead of getting it for free as in the software as a service model. The client will also have operating costs to run the software on their own server system. In this model, the software developers are economically incentivised to try and get as many new customers as possible, and they have less commitment to already existing customers than in the software as a service model.

There are many similarities and differences between the software development life cycle and Agile. The software development life cycle is a system that involves working through the software layer by layer, coding the whole project all at once. First you define the problem, then you analyze, then you design, etc. In Agile software development, you work through the software, completing one user story (functional chunk of the program) at a time. First you could create a working navigation, then create a working home page, etc. With Agile, at each step the program is functional, although it may not have all the desired features at that time. Agile also allows you to revise the program as you go, so that you can stay up to date with the clients wishes. Both Agile and the SDLC are a step-by-step method of effectively programming software, and both have their uses in certain situations.

I believe that Agile would be most effective in providing software as a service, when providing software as a service you need to be able to adapt to the needs of your clients and build on top of existing code. To quote the Project Manager, "Our system isn't something that we just design, program, and then ship out, with users having it forever in that initial state." Software as a service isn't static, and Agile's key concept of iterative development allows programmers to constantly revisit and revise their software.

I think that I would be a good candidate for the role of a senior project manager. I think that my skills in managing my time and managing people would be helpful for this job. Time management would be an important skill for this role, as you have a lot that you must get done, and you would have to balance your responsibilities. Managing people would be essential, as a senior project manager would have to communicate with stakeholders, the development team, and clients. I think that I would struggle with writing and producing detailed technical documentation, as literary skills aren't my strongest. I may also struggle with making important decisions, for example, picking which features to implement, and which to delay.