

Based on my findings, there isn't a single right answer as to which methodology is best. There are too many variables that go into an application that makes using a single methodology seem to be a poor choice. Instead we should use a hybrid of the two, one that utilizes each methodologies best features. Typically when an application is first being developed it would be of best interest to figure which features are definitely needed to release the application to the public. For the first release it seems it would be beneficial to use the waterfall methodology. This way you can stay on schedule and release a usable product to the public. Once released it seems that the agile methodology would be best used since you are now getting user feedback on the product. This way you could use the feedback and develop in iterations based on what the user wants most.

After all of the research was done I realized that the development process is never truly done. If you want a good application you need to constantly update it and stay with what is most needed. Because of this fact the waterfall methodology would be a poor decision for updates and new features. The scope of what people want is ever changing and will never be finished. Look at Facebook for example, the way Facebook is used now has changed so drastically from where it was when it was first developed because of the ever changing needs and wants of its users. Because of this fact the waterfall methodology once the initial release would no longer be cost effective and would lead to many features being obsolete or unneeded after the long timeframe of a waterfall model.

Another issue I noticed was that agile wouldn't be very useful for the initial development. While it could be used it would lead to an incomplete feel on first release and as we all know the first impression can make or break an application. Since the agile model develops in iterations users may only see a part of what the application is supposed to offer upon first release. This would lead to unhappy users and less followers leading to an application that would not be very successful. I know from personal experience that if an application feels incomplete upon first opening it I am more likely to delete it and never use it again.

This lead me to think that some sort of hybrid way of developing would lead to the best applications and the most users. If you could use a waterfall model or maybe a dumbed down waterfall model to develop the initial release, this would lead to a more complete initial release. Having a complete feeling release would essentially get you more users and make them want to use your application. After your initial release I would use the agile model. This is because you now have a good base of users ones that are happy and more willing to leave a review on what they want from the application, which plays great into an agile model. Knowing what your users want, you could develop in iterations based on the most sought after improvements. This would lead to even happier users that are willing to recommend your app to a larger user base.

Agile Methodology Agile Software Development. (2008, October 23). Retrieved April 11, 2016, from <http://agilemethodology.org/>

What is agile project management? - Definition from WhatIs.com. (2011, August). Retrieved April 11, 2016, from <http://searchcio.techtarget.com/definition/Agile-project-management>

Waterfall vs. Agile. (n.d.). Retrieved April 11, 2016, from <http://www.onedesk.com/waterfall-vs-agile/>