Systems of development, where as the system

Business, Management

Systems development life cycleSystem development life cycle is a series of phases to create hardware/software to meet a specific set of needs (Alwan, 2015). There are 6 major phases of the system development life cycle; Planning, Analysis, design, implementation, integration, and maintenance (Alwan, 2015). Planning requires research and analysis of the current problems, needs, and wants for the software in order to decide what it is they are going to create. Analysis goes on to test the technology involved and if the system will work for them.

Once these things are known, the software can be designed and the elements can be put together. Implementation of the system comes after these phases where code can be written, installed, and they receive training on how to use it. Integration is where testing is done by users of the software. Maintenance is then done periodically to improve performance and extend the relevant life of the software. An appropriate example of when to implement systems development would be a restaurant looking for new software to integrate inventory management with point of sales computers used by staff. A company will then go through the life cycle phases to develop the new software for the restaurant.

Agile developmentAgile development method, unlike the stricter process of system development life cycle, is iterative (Agile Alliance, 2011). An example of when an organization might want to use Agile development would be a restaurant looking for automated data gathering from their customers to identify trends in key performance indicators. Here, the requirements may change or new ones identified as the process of development progresses, creating a system that evolves with the creation process. The core of Agile

development is values and principals, as outlined the The Manifesto (Agile Alliance, 2011); 1. Customer satisfaction by early and continuous delivery of valuable software2. Welcome changing requirements, even in late development3.

Working software is delivered frequently4. Close, daily cooperation between business people and developers5. Projects are built around motivated individuals, who should be trusted6. Face-to-face conversation is the best form of communication7. Working software is the primary measure of progress8.

Sustainable development, able to maintain a constant pace9. Continuous attention to technical excellence and good design10. Simplicity is essential11. Best architectures, requirements, and designs emerge from self-organizing teams12. Regularly, the team reflects on how to become more effective, and adjusts accordingly Agile development accepts and embraces the changes in the process of development, where as the system development lifecycle does not and requires upfront research and design. Agile goes with the flow, and works incrementally.

However, a downfall to Agile is time and money. With system development a company can specifically outline developer labor hours, details, and contract price. Agile cant do that because the process changes and is heavily involved in each new and changing development. An organization may not have the time needed to work with developers, and the cost can significantly increase as more time is spent on the software development.