

June 3, 2018

Course: CIS570 – Business Intelligence
Name: Robert Palumbo
Assignment: Reading Discussions – Week2-Session1
Due Date: Sunday, June 3 @ 11:59pm

Question 1. In your opinion what data warehouse architecture is best? Why?

It is my opinion that there is no *best* DW architecture. They have all been designed to ideally fulfill unique situations and requirements.

Generalized querying by end-users would best be served by a traditional RDBMS which is likely the case for most organizations that are not in the business of providing for example cloud services such as AWS, Microsoft etc. As we know this type is best suited for OLTP. If the business is not greatly concerned about overall performance then this will serve well. This type is well understood with considerable industry expertise available.

For organizations that require large amount of data aggregation for reporting purposes the CODBMS is the architecture of choice. Being column-based (vs. row-based as in RDBMS) it is optimized for aggregation, which is the accumulation or summarization of column data. The data itself becomes the primary key which columns are then indexed on. This is extremely efficient and as we have learned is more appropriate for OLAP. This type is I would argue is on the non-standard side of the house likely involving a bit more specific expertise.

For organizations that are involved in heavily complex analytical processing where performance is everything, then the In-Memory CODBMS is the choice. This option places *all* the data in physical memory. Since the entire database resides in memory the performance bottlenecks that are associated with the other methods are basically eliminated. This type is incredibly fast for data processing and retrieval; however, the tradeoff comes at the expense of building out a system with enough RAM enough to store the entire database. RAM is costlier than disk space and thus it will simply cost more to build out such a system.

In the general sense I would argue that for the small to mid-size businesses, a traditional RDBMS is *best* as for all intents and purposes as I feel it would serve the basic needs of the organization in terms of costs and performance. I believe that it is the best understood design and that translates over to costs for supporting it etc., but more than that, the organizations should be able to achieve their objectives without having to rely on more sophisticated alternatives.

Question 2. What is Scrum development? Discuss the key arguments in support of this approach to developing software.

Scrum development is a formal methodology that falls under the more encompassing *Agile Development* philosophy. Whereas Agile describes an approach or framework for process management, Scrum provides a formal set of steps and rules to follow the software development life cycle.

Both approaches arose out of the need to redefine the SDLC due to myriad project failures that used a more traditional approach of software development in which the project plan followed a very systematic and sequential set of steps that was not designed to accommodate change during the process. Issues that were found at the very late stages would often require having to go back and redesign component of the systems, sometimes the entire architecture, which would ripple back through other system components. A very costly undertaking for sure.

Scrum is one technique that addresses this issue by employing an iterative model for development. It is a team-based approach which includes a single scrum master, product owner, and the scrum team. The product owner guides the vision of what is being developed, the scrum master provides overall management of the scrum process itself, and the scrum team which is usually a small group (3-5 members) charged with performing the actual development work.

The team creates *user stories* which are used to drive the tasks that must be completed during the process. A *product backlog* is created from these stories which is a list of desired for the product. *Project* management is broken down into measure able units of time called *sprints*, typically 2 weeks in duration. *Sprint planning* is used organize the backlog with the top priority items from which to choose to work on during the sprint. The team then selects from the list the work they feel they can completed during the sprint interval. Another key component is the *daily standup* meeting whereby every day during the sprint members inform each other of their goals for the day and any issues they have encountered. This is critical because it facilitates being able to collaborate any issues and addressing them as soon as possible.

The advantages of Scrum is that is an iterative process that break the development process down into small, manageable units of work. It allows for a more collaborative environment that accommodates change that can be addressed at any time during the process. By detecting and resolving issues as soon as they are encountered this can greatly reduce the overall costs of the project. This in turn improves the overall quality and robustness of the product. Likewise, with daily standup meetings, everyone on the team is aware of each other's status which again leads to improved product quality as the team can identify potential issues and address them as soon as possible.

<https://www.scrumalliance.org/>
<https://www.agilealliance.org/>