­­­Roger Baumbach II

Doctor Pomykalski

Info Systems: Database Analysis

November 25th, 2013

CS3: Balmoral Group Final Case Study

Section 1: In the field of information systems, most if not all systems analysts and technical analysts use a number of tools called models, which helps to structure the design path of a particular informational systems project. A model is a [simplified] representation of either reality or vision (Pomykalski PowerPoint 07). These models are also part of what is called the Systems Development Life Cycle (SDLC). The four stages in this process are the planning stage, analysis stage, design stage, and implementation stage. For this case study, there are five models within a business venture that need to match the Balmoral Group’s approach.

The first model, the Systems Service Request, is used when a person in the business entity either sees an opportunity or problem. This model comes before the Analysis Phase and is used to make a suggestion to the systems analysts department for further study or initiative; this is known as the bottom up approach. At this point, people who are in contact with the system on a daily basis can make a suggestion for the system. This is opposed to the top down approach, which occurs when an executive in a business entity makes a request for a system that would help the business as a whole. This is similar to Ackerman’s approach in that, Ackerman, the CEO of Balmoral, saw an opportunity to create a system that would help Balmoral as a whole. The second model that will be used to study the Balmoral Group’s approach is the Statement of Work. This the final model in the planning phase and the Statement of Work is used to make a formal agreement between the systems development team and the client to define the specifics and outcomes of the upcoming development process. The third model that will be used is the Functional Requirements model: it is an activity to determine inputs, outputs, and the behavior of the system. It is part of the analysis phase in which requirements will be determined. The fourth model is the Entity-Relationship Diagram or ERD for future reference and it is part of the planning phase. An ERD helps as a model to show the flow of data between its outside entities and the database. The fifth and last model that will be used is the Database Application. The database application is the actual deliverable from the Systems Development Life Cycle, or SDLC, which has three components to it that need to be further, analyzed; they are forms, reports, and tables. The database Application is the actual model or deliverable that is used to store data, organize, and then transpose it to an output that is organized and more comprehensible than the original data.

Section 2: In 2005, there are about 500 industry analysis worldwide (Mallach, 25). On average there are about 100 analysts per firm and up to 500 or 600 in some. In the early part of 1999, systems analysts were regularly quoted in technical and business periodicals with their recommendations on the current business problems and structures. In terms of tracking all of these analysts and anything that they may recommend, it is a nightmare in terms of organization. One man, Lawrence Ackerman, CEO of Balmoral group, decided to undertake the task of recording all of these names and their quotes could “give insight into their influential areas” (“A Database Project in a Small Company” 26). If he was able to collect and organize of all these quotes from analysts, the data alone would be invaluable to any business entity of the time. So Balmoral’s goal was to compile all the quotes of these analysts into one location.

Section 3: Lawrence Ackerman’s goal was to make a “database” of all the quotes and views of the analysts so that he and his consumers could utilize this information and he could market it. The chief customer for this product Ackerman had been designing was the Information Technology community that would like insight into the current views and ideas of experts in their respective fields.

Section 4: The Database got its beginnings in 1999 when Ackerman had been sitting in the Chicago O’Hare airport, reading *InfoWorld*. While reading *InfoWorld,* Ackerman had a thought that if he recorded the thoughts and concerns of the professionals that are quoted in this Magazine, he might get insight into their specific field and it would help to provide a useful advantage. He pulled out his laptop and began the database by recording a set of information about each analyst in excel:

* Analyst Name
* Job Title
* Firm
* Location (city; also state if U.S., country if not U.S.)
* Topic of Article (a few words indicating what the article overall was about)
* Article Title
* Publication Name
* Date of Issue, Volume, Number (as applicable)
* Writer(s)
* Point(s) Made (summary of what the analyst was quoted as saying)
* Vendor(s) Mentioned
* Entered by (initials, for the possibility that others might enter quotes at some time)
* Date Entered

After a few months, Ackerman presented this database design to a committee of his fellow colleagues. The database was in a flat-file structure and it did not have any retrieval facilities other than text-search capability of its underlying package of Excel. It was also missing many features that were missing from the original draft such as attitude towards a vendor. Ackerman at this point decided that it was time to form an informal version of the database and a prototype was formed. Ackerman and his colleagues formed an actual database using FileMaker Pro 5. Ackerman made a loosely structured form of the database, and imported most of the data from his excel spreadsheet.

The database itself was done at this point and ready to be updated with new entries. Though, there were many flaws with the design of this database: Only one user could edit the database at a time and it was very minimal in features for the end user. Ackerman also had a cluttered database: it had multiple entries from the same analyst and had no simplification of data. Names for some analysts were not standardized: some may be misspelled, others may have been only a first or last name, and others were abbreviated. Not all things matched up with the correct data in another table.

As time went on, the database was making a third of the total business’s revenue and was still very defective. He implemented it from scratch with no pre thought out plan. By 2001, the first version of the database had been released and it came with the user side interface, which gave the user an easy to manage search function. Following this, Ackerman had organized his data into more tables making it less bulky and simplifying repeated data; which could have come originally if he would have gone into the project with a pre-thought notion of how the tables and data should have correlated, otherwise done by an ERD. Later on, Ackerman then produced a User Manual, by himself, which grew from 20 to 35 pages and reduced the number of customer issues. In 2002, the database received two new features that first allow searches by region and coverage. The next was a revamp of the searching feature for accessing analyst information. The last feature that would be implemented was a table that would contain a company’s notes on an analyst that would not disappear with each new update of the database. In the end, the database underwent two final changes before a bigger firm bought it. They eventually put the database online which appeased the customers who were distraught because they had to wait 6 weeks to receive a new version. The last change made, the new owner at this point, Isabelle Oliviera moved to have the entire database moved to SQL for the purpose of design.

Section 5: In context and in relevance to the Systems Development Life Cycle, the Balmoral group was very careless in their development of the system. From the beginning, the system had a large potential to become a staple in the business world and turn a large profit but, the database was designed very poorly. In relevance to the first and most important model to the system, the Systems Service Request, there was none. It caused so many problems for Ackerman in his company because there was no solid plan through the entire production of the system. The lack of planning resulted in increased development time and reduced efficiency in the database design and production. What Ackerman did have in starting his design was a prototype of the database. The prototype itself was flawed, clunky, bulky, and had no useful features other than those that are standard with Excel. In retrospect, if Ackerman had raised this concern formally with his employees, they could have drafted a Systems Service Request, and they could have come up with a better design for data collection and organizational columns. They would have been able to map out the entirety of the database beforehand, and in addition to the better data organization, he could have set up a team and worked with them to move towards a standardized database. In the end, he may have had a more efficient and reliable database in a shorter amount of time.

The next model in question is the Statement of Work. The Statement of work, as mentioned before, is a document summarizing the work to be done and the agreement between client and database designer. During the production of the model, a Statement of Work was not present. There wasn’t a single hint or formal documentation of the database’s intents or purposes between the creators, the Balmoral group, and their end users. Ackerman took the approach of starting a prototype at the O’Hare airport in an excel chart, and then when testing it, he brought it to his coworkers to critique. When marketing it, he was still in the beta process of its development and crowd sourcing the consumers to see what features he should implement. If he would have taken the approach of formally drafting a plan, meeting with a consumer one-on-one, and developing the database with fellow staff members, he might have been able to turn the product around and market it as a monopoly. Without the Statement of Work, the Balmoral group had little to no direction in production. They had no written goals, objectives, or schedules to abide to, so they produced in an untimely and costly manner in which the system took too much time.

The functional requirements were a subject lightly touched upon in the production of this system. The Balmoral group did not really come into the production of a plan of what the database should do, but only what kind of data should it track. In actuality, it would be hard for Ackerman or anyone to create a database like he did without first considering the Functional Requirements. Ackerman first drafted a list of required inputs to the system before he even began collecting entries for data. After compiling his list he decided how the system would work which was very simplistic in design. He or another analyst would compile the data and place it into the excel sheet and distribute that excel sheet to those who needed the data. After this had gone on for some time and with much crowd sourcing, the project evolved and a database was formed. When the database had been formed, Ackerman and his employees drafted a solid list of required data for each entry that is made, and the outputs for the system were planned out. The main change that occurred was when the database application was created in FileMaker Pro. At this point, Ackerman had created a solid design for the end user to access, decide on data, and receive a list of results that they wished to receive in a report that would be compiled. The other big improvement was the design of multiple tables in the system which meant that multiples of the same data were removed and that all the data wasn’t located on a single chart. These improvements were handy because they rid the system of the hindrance of not having a solid search feature other than Microsoft Excel’s built in word search function. The real problem though, is how do you really test a system though without the requirements? The idea of testing something means that one is attempting to see that the object, program, or thing does what it was intended to do. How could one test something if they don’t really know what it is to do? Yes, Ackerman had a plan as to what he wanted it to do, but how was it supposed to do what it was intended to?

The fourth model, the ERD or Entity Relationship Diagram, wasn’t used at all in the design process. The only time in which the case study of the Balmoral group mentioned an ERD was when they made one to represent what the system was doing currently. The Balmoral group never even attempted a model in the beginning stages of the process. They began with only the criticism of his fellow colleagues and the feedback of his clientele from the databases initial testing. Ackerman put everything into one single model and didn’t even attempt to combine duplicates or conserve space in his model. Every entry was a large cache of data that may have had multiple values or inconsistent names and values. By the end of the article, the Balmoral group had separated their one table into many smaller tables in which duplicate entries had been condensed and/or partially removed. If the database had begun with an ERD, going into the actual modeling process, the group would have condensed data early on and made their job so much easier and more efficient, even if they had been using excel at that time. The model was used late in the process after the fact and it resulted in more wasted time and the efficiency of the company dropped.

The last model mentioned, the Database Application, in reference to tables, forms, and reports; I have mentioned multiple times before. From studying this article, I have learned that there has always been a database application in use. Since the very beginning, Ackerman started with a model despite the fact that it was often just a prototype. Ackerman never established a system for forms and reports, or inputs and output documents for the system. He simply added the data to the excel sheet as it needed to be added then redistributed it to his clientele. He also had a solid system of how to plan out all of his tables in a consistent manner. As time went on, a solid system for reports had been developed when the system moved to FileMaker. This enabled Balmoral’s clients to choose specific data requirements and the system would then compile and output the data to the user. The tables also evolved to encompass more tables that were less broad and more pinpoint by repeating similar data and relating tables through primary keys. The article never stated if the database had come up with a system for entering data through a system of forms. The company still had to manually input data themselves but at least more than one person at this point could input data to the chart. There was also another flaw where the database did not have a feature to output and organize data in a report format.

Section 7: In retrospect, I find that the whole approach to designing this system was convoluted and unprofessional. I think in the beginning, Ackerman started off well. He jotted down a few notes and example quotes, formed data subjects that had to be collected upon each new entry, and brought up his idea in a meeting with colleagues. Brainstorming was a great idea, but this is where the business had gone wrong. The Balmoral group, at this point, should have begun drafting up a solid plan of action for this system. If I had been in charge of the project, I would have begun with a plan immediately. After making a solid plan, I would have tried to condense my data into the most basic of structures and tables that would have resulted in the minimalist amount of input, and data entry, in which would need to be done. Then, I would have drafted an ERD model to show the relations among data in an attempt to normalize it. After completing this task, I would have moved to have my business begin an approach to getting a solid structure for inputting data quickly and efficiently. In addition to the input features that would increase efficiency, I would move to have a system of generating reports completed. This would allow my clients to choose what data they wanted, and then it would return to them the data they requested in a neat and organized format. From this point, my business and I could beta test the product for a few weeks or months to see what features would be good to implement and what are extraneous. I would then market my product professionally and have it on an in-house server that people could remotely use.

Section 8: In conclusion, I found some things of interest in this case study. Before reading this article, taking this class, and learning the process for designing a system, I used to jump head long into my programs personally as a Computer Science Major and taking away from this, I realized without a solid plan, my entire project may come crashing in on itself. Another thing I take away from this is the importance of deciding on entities in a system and how to structure my data. I realized that a majority of the time I work on something, if I don’t condense my data and things into similar tables and cut out repeating entries, I could end up eating up my data storage space and having to spend more money on hardware, costing my business more of our profits. The last thing I learned is the importance of testing a project before releasing it. It comes off very unprofessional that Ackerman released a product like he did and had to use his clientele to essentially beta test the product. He then built an actual database from the use of his paying customers. This is unprofessional for one and it also he had many different clientele that each may have wanted to database to perform a certain task and it didn’t coincide with the idea of having a set path or goal for the system.