Journal of Case Studies **May 2017, Vol. 35, No. 1, p. 50-56**
www.sfcrjcs.org **ISSN 2162-3171**

Developing a Data-Driven Application with Damaged Data and Anomalies

Edward V. Weber, Millikin University

Introduction

Ed's stress level had completely shot through the roof. As a computer consultant who had owned and operated his own business for nearly twenty years, he had encountered these types of problems numerous times in his own extensive career. But this time, as a newly hired Assistant Professor of Information Systems for Millikin University, things were completely different. Ed began this project fully confident that this data-driven application would be a perfect fit for his Systems Analysis and Design students as well as his Web and Mobile Programming students. Then when the project was first approved by the President and his cabinet and the students started really getting into the design work, Ed had felt really good about the wonderful Performance Learning opportunities for all of the students.

But then it started. As the students continued their analysis and design work, they found themselves running head-first into one data anomaly after another. Quickly, the students discovered that the real-world project realities did not match their textbook examples. One-by-one, these data anomalies kept showing up like uninvited party crashers who everyone knew would simply never leave.

Ed worried if his students would be able to handle these new realities and setbacks. He worried if they would be able to make the necessary leaps in critical thinking to move forward from their initial designs and assumptions. He worried if they would be able to come up with a feasible solution that addressed all of the projects needs but also handled all of the data anomalies they had discovered. As a Performance Learning project, Ed knew that the students fully understood how the success or failure of their project rested completely on their own shoulders. But this knowledge gave little comfort to Ed. Ed worried and wondered to himself, "What will the students come up with to deal with these issues and will it be enough for the project to succeed?"

Background

Ed sat pensively sipping his coffee and recounting all of the steps that had led him to this place. It had all started back in November when he was interviewing for his new professorship at Millikin University. Ed was drawn to Millikin in a large part due to their commitment to Performance Learning. At Millikin, Performance Learning meant that students learned and applied their knowledge and skills while working on real-world projects, with real-world stakeholders, and real-world risks and rewards – not just classroom exercises and simulations. Ed's extensive background in information systems and as a business owner and computer

consultant for over 20 years brought the real-world experiences of information systems into the classroom. All together, the interview committee, the Dean and Ed himself had recognized that together, they were a very good match.

It was during a follow-up interview with the Dean that Ed had come across the idea for his first Performance Learning project. Ed had mentioned that while he was investigating background on Millikin University, he was unable to find any kind of organizational chart and, with all of the people he had met during the interview process, he was hoping to see an org chart that would show him some of the hierarchy within the University. When the Dean got finished chuckling to herself and saw the puzzled look on Ed's face, she shared with him why she found his request so amusing.

"Coming from business and industry myself, I know exactly what you're looking for, but I'm afraid you won't find it. In fact, you may find a number of things that aren't done quite the same in academia as you may be used to from your own background and experiences", she said. Ed learned that Millikin University had no formalized organizational chart with which new hires (faculty or staff) or new students could discover the various members of the University family that made up the organization.

Ed continued to think about how there was no formal org chart for Millikin University and how having a searchable online version of this would be such a huge benefit to newly hired faculty, staff, and even the students themselves. Ed continued to mull the possibilities over in his mind and then, during the Summer before the beginning of the new school year, Ed met with various stakeholders to discuss this opportunity as a possible Performance Learning project.

Ed laid out the case for the President, his cabinet, and the other affected stakeholders such as the IT department and the HR department. The goal for this Performance Learning project was to utilize AGILE methodology to allow the Analysis and Design class to work alongside the Web Programming class to create a graphical organizational chart for the University. Ed explained how AGILE methodology provides iterative cycles of analysis, design, development, testing, review, and implementation and how the two classes would work together on the same project through these multiple cycles of discovery and development.

Ed stressed to the stakeholders and the decision makers that the real beauty of this type of project was that it was going to be a "data-driven" application. With a data-driven application there was no need for new data entry, data validation, or for the storing of new data. The assumption is that the data has already been validated at the time it was captured.

Ed explained that data-driven applications derive their primary functionality (navigation, presentation, interactions, etc.) from the very data that is being represented. Therefore, data-driven applications were ideal for exploring and presenting data in a meaningful way to the end users. A data-driven application would be smaller in scope and more manageable for students to deliver because the focus would be on retrieving and using existing data.

With this data-driven application, the only significant effort and investment for the whole project would come from the students themselves in the Systems Analysis and Design class and the Web

and Mobile Programming class. There would be only minimal effort from the IT department and that effort would be fully minimized since Ed himself would be coordinating and managing the implementation. Also, since this application was data driven, the existing HR data would be used and there would be no additional procedures or activities for the HR department beyond what they were already doing.

Ed sipped his coffee and remembered that initial excitement when he learned that the President and his cabinet had approved his Performance Learning project request. It wasn't very long at all before that excitement gave way to concern and then to worry.

Project Kickoff

When the project started and the students in the two classes began to discuss how they would approach their analysis and design, the students knew that they would need to have access to the data that was going to drive this application. Since the students understood that the intention was to have the application use the existing data within the organization's HR database, their reasoned assumptions about the data were very simple: First, every person on staff at Millikin had a unique record identifier to allow for the individual to be paid for her/his services to the University. This meant that each individual had a unique record in the system.

Second, through discussions with the HR department, the students learned that by University policy, each individual received an annual evaluation from her or his supervisor. This meant that there should be a unique identifier on each personnel record which identified the person responsible (supervisor) for each individual. Third, each person had a job title and a department to which they were assigned. And, each person had a University-issued ID badge which included her/his picture. *In theory*, the students reasoned, this should have been sufficient for the classes to develop an organizational hierarchy chart which would be completely data-driven. But, as they quickly discovered, that is where the theory ran full steam ahead into that brick wall known as reality.

In order to continue their analysis and design, the students requested that Ed provide them with the means to access the data. From the earliest Analysis and Design meetings, however, it was determined that, for security reasons, it would be better to not attempt to access the Human Resources database directly. This would have exposed both the application, and by extension, the University to unnecessary risk.

The students learned that many organizations create a directory of commonly used data elements, external of the raw Human Resources database, and store that data in a repository that is accessed via the Lightweight Directory Access Protocol (LDAP.) As noted by the authors at LDAP.com, for many developers who "... decide to use LDAP to interact with users, groups, and application data, that decision may be significantly influenced by an LDAP environment that already exists for other purposes." (LDAP.com, 2015)

In conversations with the IT staff of the University, the students determined that all of the critical fields which were required to drive this data-driven application should be available via the University's implementation of LDAP. The LDAP repository had all of the desired fields: first

name, last name, title, department, location, email, phone, and, most importantly, the unique employee identifier and employee's supervisor unique identifier. Additionally, using the employee identifier, the students could use an existing Application Program Interface (API) to extract the employees' pictures for inclusion in this web application.

As an integral component of this Performance Learning project, the students were required to sign a Non-Disclosure Agreement (NDA) which would govern the students' actions regarding the personnel data that they would need to access in order to complete their work on this project. (Appendix A) Through the use of this NDA, the students quickly recognized the seriousness of their responsibilities and the consequences of the tasks they were undertaking.

Analysis and Design Discovery

Once all of the NDAs were signed and a read-only view into the LDAP was made available to these classes, the analysis and discovery continued as the students began to verify their project assumptions. Very quickly numerous data anomalies were discovered:

- Some employees appeared to report to a nonexistent boss (which could be explained if the position was vacant at that time…)
- Some employees appeared to report to a boss in one department although the individual's own department was listed as something else.
- The title of some individuals did not reflect their current title.
- Some student and alumni records were showing up as being active employees of the University.
- There were inconsistencies in the phone numbers (e.g. some phone numbers had area codes, some did not, some had exchanges, some did not, some had extensions, some did not, etc.)
- There were inconsistencies in the locations (e.g. some had full addresses, some had building names only, some had room numbers, some did not, etc.)
- There were inconsistencies in the titles and departments with abbreviations and contractions appearing haphazardly.

External from the data (i.e. from direct experience) the class identified that some individuals actually worked for more than one department. But upon inspection of those individual LDAP records, it was determined that those employees only appeared to report to a single person (i.e. only one supervisor in only one of the two departments.)

During one analysis session, a student pointed out that she recognized her roommate as being listed in the LDAP as an employee of the University. The students discovered that some students were, in fact, direct members of the payroll and could be categorized as University employees. However, it was likewise determined that other students were listed as employees in the LDAP but this was only in conjunction with their work-study statuses. This was one of the many times that Ed felt that sick feeling in the pit of his stomach as he now recognized he would have to introduce his students to the ramifications of the FERPA regulations.

Under the Federal Education Rights and Privacy Act (FERPA) regulations, information related to

students that corresponds to their financial or work-study statuses had to remain protected and confidential. This meant that those student records could not be included in this new org chart and must be excluded from presentation. The students discovered much to their dismay that there was no data in the LDAP to indicate whether a student employee was a work-study student versus a 'regular' employee... only that they were a student. This 'missing' data was one of the many anomalies the students had to contend with.

But the largest data anomaly that the students discovered which would effect the overall design of this data-driven application had to do with the concept of 'departmental hierarchy.' After numerous meetings and exhaustive research, the students concluded that *nowhere, in any of the University's systems* was there a systematic, data representation of the departmental hierarchy for Millikin! As a result, according to the data, it was equally plausible that the janitorial department could have reported directly to the President's office or the Provost's office or the Athletic department or anywhere at all for that matter!

It was about this time that Ed began to wonder if his hair was actually turning more grey or just falling out completely. He remembered that the President and his cabinet had signed off on pursuing this Performance Learning project in large part because of its design as a data-driven application. By the nature of a data-driven application, there should be no need for extensive data entry or data maintenance above and beyond what was already being done in the institution. The director of human resources herself had specifically cautioned that she and her staff of only two employees were not prepared to do extensive additional data maintenance in order to support both the existing HR records as well as some new organizational chart. But what would the student's come up with to solve the problem of this missing department hierarchy data?

Both the Analysis and Design class as well as the Web Programming class recognized significant problems: Analysis of the data revealed there was both missing and inconsistent data! This was supposed to be a relatively easy system to develop – a data-driven application – which meant there were no complex data entry forms and validation rules; no complex database record manipulation and storage… It was supposed to be an easy but powerful application!

Scope Creep Appeared Inevitable

While the original scope of the project was well defined and the project scope was authorized for continued development, as these data anomalies revealed themselves, it became apparent that scope creep was on the horizon. Kahn defines scope creep as follows:

> *Scope creep is a term used to describe unauthorized scope changes. Unauthorized changes may creep into project scope as a result of verbal instructions, e-mail instructions, written instructions that have been issued without realizing the magnitude of change, etc.* (Kahn, 2006)

The students were faced with identifying and analyzing which aspects of the original scope of the project would need to change and what implications those changes would have on the overall project deliverables.

The value of a data-driven application is derived from the successful systematic analysis and understanding of the underlying data. Data-driven applications concepts are synonymous with data-mining concepts in this respect:

> *The need for knowledge discovery from real-world, low-quality data becomes not just overwhelming, but also compelling. As a result, issues related to data quality have become more and more crucial and have consumed a majority of the time and budget of data mining projects.* (Zhu et. al., 2007)

If the data-driven application is expected to perform flawlessly within the construct of the underlying data, low-quality data will inevitably become a source of potential scope creep as new data anomalies become exposed.

Conclusion

Ed was very proud of his students and the way that they had identified all of the issues they were facing. The students in these two classes fully understood all of the promises of a data-driven application. They also understood how valuable this new application would be to the Millikin community and they recognized the support for this project which was given by the President and his cabinet. However, they also understood that a data-driven application's success is predicated on accurate and complete data. The scope of this project was predetermined based upon the assumption that clean, accurate, and complete data were available. With the information determined through the analysis of the existing data, the students were very concerned about "scope creep" of this Performance Learning project.

With growing apprehension, the students asked themselves several important questions: Could they successfully build a data-driven application in the absence of complete and accurate data? If so, what adjustments did they need to make? Also, was there anything that could be done to support or alleviate the *underlying* issues which caused all of the data anomalies that they had encountered? Would they need to further involve the HR or the IT staff and what would that mean for the scope of their project? Would their solution be able to handle changes in the data in the future?

When the students turned to ask Ed these very same questions, he gently but firmly reminded them that this was *their* Performance Learning project. *They* were the ones who needed to come up with appropriate solution options that they could all agree upon and that they could fully develop within the remaining time of the semester. But deep down, Ed was camouflaging the mounting stress that was churning in the pit of his stomach. Ed couldn't help but worry and wonder what solution the students would come up with to fully meet the needs of this data-driven org chart without incurring excessive scope creep?

References

Khan, Asadullah. "Project scope management." Cost engineering 48.6 (2006): 12-16.

Why Use LDAP? (2015). Retrieved July 18, 2016, from https://www.ldap.com/why-use-ldap

Zhu, Xingquan, et al. "Editorial: Special issue on mining low-quality data." Knowledge and Information Systems 11.2 (2007): 131-136.