Spring 6-10-2017

EXPLORING HOW DIFFERENT PROJECT MANAGEMENT METHODOLOGIES IMPACT DATA SCIENCE STUDENTS

Jeffrey Saltz
Syracuse University, jsaltz@syr.edu

Robert Heckman
Syracuse University, rheckman@syr.edu

Ivan Shamshurin
Syracuse University, ishamshu@syr.edu

EXPLORING HOW DIFFERENT PROJECT MANAGEMENT METHODOLOGIES IMPACT DATA SCIENCE STUDENTS

Research in Progress

Saltz, Jeffrey S, Syracuse University, United States, jsaltz@syr.edu

Heckman, Robert, Syracuse University, United States, rheckman@syr.edu

Shamshurin, Ivan, Syracuse University, United States, ishamshu@syr.edu

Abstract

This paper reports on a controlled experiment comparing different approaches on how to guide students through a semester long data science project. Four different methodologies, ranging from a traditional "just assign some intermediate milestones" to other more agile methodologies, are compared. The results of the experiment shows that the project methodology used in the classroom made a significant difference in student outcomes. Surprisingly, an Agile Kanban approach was found to be much more effective than an Agile Scrum methodology, which was not one of the leading ap-proaches.

Keywords: Data Science Education, Big Data Education, Project Management, Agile Development.

1 Introduction

Data science is an emerging discipline that integrates concepts across a range of fields, including computer science, information systems, software engineering and statistics. Due to student demand and the growing number of data science programs (O'Neil, 2014), there has been an increase in the number of data courses offered at Colleges and Universities. While the number of data science courses continues to increase, there has been little research on how to most effectively teach a data science course.

This paper explores one aspect of teaching data science, namely how to best guide students during a group data science project. In fact, while the use of projects within a data science course is fairly common, there has been little research on how to effectively guide students through these course projects. Specifically, this paper focuses on the impact of different approaches to student oversight and guidance, using a controlled classroom-based experiment. In this experiment, we evaluate student teams using different methodologies to coordinate and execute the project. Our research aims to understand if one process is better than the others (with respect to what is the best methodology a group of students should use to do a data science project) by focusing on the following research question:

> Are student outcomes different, based on the type of process methodology that is used for a student project?

The rest of the paper is structured as follows. Section 2 contains a review of related research. Section 3 describes the methodology for our empirical study. Section 4 presents the results from the study, and finally, section 5 discusses conclusions and limitations from the study.

2 Literature Review

Below, we summarize recent research related to data science education, methodologies of teams working on data science projects and the more general area of evaluating software projects (as a proxy for previous research evaluating student data science projects). Perhaps because it is a new domain, beyond what is reported below, there has been little focus on how a group of students can best learn data science via a course project nor if the methodology used can impact the level of student enthusiasm or student outcomes. More broadly, there has also not been significant discussion with respect to the challenges that students might encounter when they are doing a data science project.

2.1 Data Science Education

There has been a bit of research published on the slightly more general topic of data science education. For example, some have focused on designing a data science curriculum (Ramamurthy, 2016, Anderson et al., 2014) and others have focused on the overall design of an introductory data science course (Gil, 2014; Brunner and Kim, 2016). A bit closer to exploring how to best guide students during a project-focused data science course, Saltz & Heckman (2016) review a project-focused data science course. However, that paper was focused on the viability of using real world projects and did not address the question of how to best guide students through their semester long project.

2.2 Data Science Project Methodologies

Beyond the classroom, current descriptions of how to do data science generally adopt a task-focused approach, conveying the techniques required to analyze data. For example, Jagadish (2014) described a process that includes acquisition, information extraction and cleaning, data integration, modeling, analysis, interpretation and deployment. Interestingly, while Espinosa and Armour (2016) agreed with these typical steps, they also noted that the main challenge is task coordination.

This step-by-step data science process description described by Jadadish and others does not provide much guidance about the process a data science team should use to work together (Saltz, 2015). For example, Vanauer, Bohle and Hellingrath (2015) noted the lack of an empirically grounded data science methodology. Hence, not surprisingly, it has been observed that most data science projects are managed in an ad hoc fashion, that is, at a low level of process maturity (Bhardwaj et al., 2014). Indeed, it has been argued that projects need to focus on people, process and technology (Gao et al., 2015; Grady et al., 2014) and that task coordination is the main challenge for data projects (Espinosa and Armour, 2016).

Researchers have begun to address the need for a team-based data science process methodology via case studies to understand effective practices and success criteria (Das et al., 2015; Saltz and Shamshurin, 2015; Gao et al., 2015). Perhaps not surprisingly, it has been reported that an improved process model would result in higher quality outcomes (Mariscal et al., 2010) and at least some managers are open to improving their process methodology, but might not think of doing it unless prompted (Saltz and Shamshurin, 2015).

2.3 Evaluating Student Software Projects

Since there has been minimal research with respect to how students should work together on data science projects, examining how instructors have guided software development projects might provide some context for how to guide students in executing data science projects. Agile project management, which focuses on improving interactions and collaboration over process and tools as well as enabling teams to respond to change (Dybå & Dingsøyr, 2008), have been studied extensively. For example, Anslow and Maurer (2015) noted that teaching group based agile software development was difficult. On the other hand, Schroeder (2012) found that the agile scrum methodology was ideal for introducing software processes, and Kropp and Meier (2013) found that using agile methodologies had a positive effect on students with respect to teaching software development. Damien (2012) also described a software development course that used an agile scrum methodology. In essence, these efforts have focused on identifying best practices for defining a project, such as ensuring the project has a real world context and encouraging the regular assessment of the team (Ohland et al., 2015), but these efforts did not explore if one agile methodology used was better or worse than any other project coordination methodology, in that there was no effort to explore the strengths and weaknesses of using different methodologies within a classroom environment. More broadly, Borrego (2013) reviewed 104 articles describing engineering and computer science student team projects and noted that few of the articles discussed team effectiveness and that there is a great opportunity to address this gap.

3 Methodology

To investigate the impact of using different project management methodologies within a group project, an experiment comparing four different approaches was conducted. Since we were able to compare multiple teams working on the same project, an experiment was deemed an appropriate methodology to compare and analyze the impact of the different project management approaches, and hence, an experiment is an appropriate way to explore our research questions. Specifically, student teams in a master's level data science course worked on a semester long data science project using one of four different project management methodologies. This section describes the experiment in more detail.

3.1 Project Description

As part of the course, students were required to work on a group project, which started in the second week of the semester and continued until the end of the semester, thus lasting a total of twelve weeks. The final project was twenty-five percent of the course grade, and in general, the students were highly motivated to work on the project. Students were required to use the R programming language, a popular data science tool that is used in both industry and academia. The analysis was expected to include many typical data science techniques, such as leveraging machine learning algorithms, association rule mining and geographic information analysis.

The dataset for the project was a modified version of a real dataset of survey responses from an organization in the hotel industry. The dataset contained roughly three million responses to a customer survey and had a size of approximately 100GB. In all, there were 237 survey attributes (or columns) in the dataset, with students having access to a description of each column of the dataset. The attributes included information about the person who responded to the survey (ex. place of residence). Other attributes included the hotel (ex. location) and the actual responses to the survey from the customer who stayed at the hotel (ex. would they recommend the hotel to a friend). Note that some values in the dataset were blank. This reflected a typical 'real-life' challenge in how to handle missing values, that was due to the fact that some of the surveys asked more questions than other surveys.

The goal for each team was to identify and then answer "interesting" questions (such as understanding how customer satisfaction varied across surveys (geography, different hotels, frequent vs non-frequent guests, etc.). Note that no specific questions / goals were provided to any of the teams.

3.2 Student Participants

A total of 85 graduate students participated in the study, with all of them taking an "applied Data Science" course. 40% of the students were female and more than 75% of the students had previous IT experience. In fact, the majority of the students had two to five years of work experience, typically within the IT industry. Due to their prior work and educational experience, most of the students had experience (or knowledge) using agile and/or waterfall software development methodologies, but not within a data science context.

3.3 Experimental Conditions

All students attended the same weekly large-class lecture, which focused on providing an explanation of key data science concepts. In addition to the weekly lecture, the students were divided into four lab sections, with twenty to twenty-two students in each lab section. Each lab section also met weekly. Students selected lab sections based on what fit their schedule. Students did not know which section would use which methodology. While not randomly assigned to lab sections, all the students, across all the sections, were part of the same graduate program. All the students participating in the different of the lab sections were similar in terms of knowledge, experience and ability. We consider this a "natural" experiment, where experimental and control conditions are determined by factors outside the control of the investigators, but the process governing the exposures resembles random assignment.

Within each lab section, students were divided into teams, with four teams, of four to six students per team. The teams within each section were formed by random selection of students, which is consistent with what Ko reported, in that most studies use simple random assignment (Ko et al., 2015). The weekly lab covered practice in using hands-on data analytics as well as project time for the students.

Each lab section was taught and used a different methodology, with respect to how to work together as a team on the data science project. The methodology was explained via a mini-lecture within the lab. Students were also provided handouts and access to a process expert to ask questions about how to use their methodology. The quality of process instruction was kept constant by having the same person describe the different methodologies across the lab sections. Hence, each lab section was a different experimental condition, and each experimental condition had between twenty to twenty-two students. This is also consistent with Ko, who noted that it is reasonable to use twenty participants per condition. Below we describe the different experimental conditions.

Agile Scrum: This methodology was adapted from the agile scrum methodology that is typically used to develop software systems. Specifically, the team was instructed to do a series of "sprints" (a two-week time period during which agreed upon work was to be completed). The team collectively determined what could be done in the sprint (the two-week work effort) – with the end result being something useable at the end of the sprint. The students were further instructed that the work to be done in the sprint shouldn't change for the duration of the sprint (any thoughts and suggestions would go into the planning of the next sprint). The team was to make sure it finished all the goals of that sprint in the 2 weeks, and then meet again to jointly reflect on the sprint and determine what to do in the next sprint. More specifically, for each sprint, a "sprint planning meeting" reviewed the "sprint backlog" and then team members worked together to define the goals for the upcoming sprint.

Agile Kanban: Agile Kanban combines a set of phases to do data science (based on CRISP-DM and other recent publications) integrated with the pipeline process management from Kanban. Kanban was created for lean manufacturing, but has been adopted across a number of domains, including software development (Ahmad et al., 2013). A key aspect of this methodology is the 'Kanban board', where the work in progress can easily be seen and tracked. Specifically, the phases shown on the Kanban board included preparation (understand business context and the data), analyze (model/visualize, test/validate) and deploy (share/communicate results). Within each phase, there was defined a maximum number of work-in-progress tasks that could be "in that phase". Using this framework, the team defined a prioritized list of what to do (via high level "user stories", such as link weather data to our previously collected data). Then, based on the number of allowed simultaneous tasks at each phase, the task flowed through the defined process. Limiting the number of tasks within any one step helps to ensure the team minimizes bottlenecks and work in progress.

CRISP: Based in an industry standard, CRISP-DM (Shearer, 2000), each team followed the keys steps in a typical data project (business understanding, data understanding, data prep, modeling, evaluation and deployment). Using this framework, the team progressed through the different steps (or phases), as they deem appropriate. As needed, the team could "loop back" to a previous step (ex. more data preparation), and in general, could define milestones they thought were useful. At a minimum, a bi-weekly status update meeting was held to track status / issues.

Baseline (no defined methodology): In this condition, the students were not given any special project management process suggestions. Hence, the teams worked as they pleased, just as they would do on other team projects. Note that this was the first time this class was taught using with this project, so there was no baseline on results from previous semesters.

4 Findings

In this section we report on the findings from the experiment. In reporting the results of the experiment, a comparison of the quality of the final projects is discussed (was the overall quality of the projects different, based on the experimental condition?). This is followed by an analysis of the student reported perceptions, both qualitative and quantitative, about using their assigned methodology. Note that typical project metrics, such as a sprint burndown report (which reports on the tasks being completed) could not be used across the different experimental conditions since different project management approaches typically use different metrics.

4.1 Measuring Team Effectiveness

To measure team effectiveness, a multi-dimensional approach was used. First, quantitative data was collected via grading of the final projects. In addition, qualitative and quantitative data was collected via a post-project student questionnaire. Specifically, the questionnaire first obtained the team and section of the student, and then asked several structured questions, which provided quantitative data on topics such as would the students like to work with their team on future projects, how well the team worked together and did they find the methodology easy to use. The survey also had semi-structured questions focused on what worked well for each team. Finally, students were observed in the classroom, and these observations are integrated into our discussion of the results.

4.2 Measuring Project Results

Two experts independently evaluated each project (on a scale of 1 to 10, with 10 being an exceptional project). Across all the projects, the scores from the two experts had a correlation of 0.8, and no project had a difference (between the two reviewers) of more than one point (on the 10-point scale). The project scores within each condition were averaged across the expert reviewers.

As shown in Table 1, teams that used CRISP and Agile Kanban methodologies did better than the other two experimental conditions. In fact, there was a statistically significant difference between groups as determined by ANOVA. Specifically, using the Fisher post hoc test, Agile Scrum was statistically different from the Agile Kanban and CRISP results.

Section	**Average Score** (1 to 10; 10 is best)
Agile Scrum	6.5
Agile Kanban	7.8
CRISP	8.4
Baseline	7

Table 1: Project Results

4.3 Student Survey Responses

As shown in Table 2, at the completion of the project, via a survey, the students were asked to agree or disagree (using a 5-level Likert scale) with several statements. For example, we explored if the team members would like to work together on future projects.

Statement	**Section**			
	Scrum	Kanban	CRISP	Baseline
If it was possible, I would want to work with this team on future projects	3.4	4.2	4.3	3.8
I am very satisfied (with respect to working on this project)	3.9	4.3	4.4	4.4
This project management method was similar to how I have done previous group projects	3.3	2.5	3.3	3.6
It was complicated to use the project management method within my team	2.6	3.1	3.0	2.6

Table 2: Student Survey Responses (Average Score on 5-level Likert Scale)

Willingness to work together on future projects: If a team was highly productive but the team members never wanted to work together on future projects, that would not be a desirable outcome within many organizations. With respect to the question "I would want to work with this team on future projects", Kanban and CRISP scored the highest (with, respectively, a 4.2 and 4.3 score on a 5-level Likert scale). The lowest ranked methodology was the Agile Scrum, with an average response of 3.4, below even the Baseline methodology. Note that there was a statistically significant difference between groups as determined by ANOVA. Specifically, using the fisher post hoc test, Agile Scrum was statistically different from both Agile Kanban and CRISP methodologies. While willingness to work together could be influenced by other factors, such as student personalities, the results suggest that the process influenced the student's perceptions of the other students on their projects.

Satisfaction of individual team members: The Agile Scrum process again ranked the lowest, with a score of 3.9 (the others all had a score of 4.3 or 4.4). While interesting, there were no statistically significant differences between group means as determined by ANOVA.

Ease of Use: The Agile Kanban methodology was reported to be the most different from what the students had experienced in the past, based on the student responses to the statement "This project management method was similar to how I have done previous group projects", in which Agile Kanban methodology was much lower than the other methodologies. Note that, using ANOVA, these results were statistically significant in that Agile Kanban was statistically different from both Agile Scrum and the baseline. Our observation of student teams during the project led us to believe that an ability to easily adopt and use the process might be a key factor to consider. Hence, this was explored on our post project survey. Perhaps not surprisingly, the Kanban method appeared to be complicated for team members to use (based on the response to the statement "It was complicated to use the project management method within my team"), as was the CRISP methodology. However, due to the variability in the participant answers, none of the results were statistically different.

Perceptions of What Worked Well: In analyzing the more open-ended question of "what was working well", as shown in Table 3, the percentage of students that mentioned "team coordination" or "teamwork" was dramatically different across the different experimental conditions. In particular, 58% of the students using Agile Kanban stated that their team worked well together (without any prompt about teamwork or how the team was working together). Agile Scrum and the Baseline were much lower (19% and 15% respectively). Note that the other comments about what worked well (and the comments about what needed to be improved) typically focused on the actual project assignment (ex. "provide data at the start of the semester" or "provide more clearly defined requirements"). This last comment highlights a difference as compared to other, more typical, student projects, in that the students were provided the data set and a business champion that desired to get "knowledge from the data", but the students did not get a set of specific directions, such as which machine learning algorithm to use to gain insight into a specific hotel attribute that might have driven customer satisfaction.

Section	**% of students** mentioning "team" or "coordination"	**Representative Quotes**
Agile Scrum	19%	"We are so proud of what we have done" "Team Statistics, Group discussion"
Agile Kanban	58%	"Overall I liked the idea of the PM methodology for the project" "The team worked together efficiently."
CRISP	44%	"Collaboration and team work" "We are focused on our goals and communication has been spot on"
Baseline	15%	"The project progressed at a steady rate and completed successfully" "The team was coordinated"

Table 3: What Worked Well

5 Discussion

In response to the research question (are student outcomes different), there were two approaches that were better than the others (Agile Kanban and the CRISP methodology). Perhaps a bit surprisingly, the Agile Scrum approach was worse than the Baseline condition of no defined process. Based on the observations of the student teams (during the project updates as well as how the teams actually worked on the project), below we provide some additional context for each of the experimental conditions.

Why was Agile Kanban effective? The Agile Kanban teams used their Kanban board as a way to easily understand and explain their project status. In general, the teams had a good grasp of the client requirements. It also appears to have a low barrier to entry since even though students had not previously been exposed to a Kanban methodology, it was intuitive for them to use. Also, this approach did not require students to estimate how long each task would take – the key focus when using Kanban was only to monitor the "work in progress". For example, one team created a new "Kanban board column" to manage / balance the work done on a smaller (easier to use) dataset and how much to focus on the larger dataset. The team wanted to first work on the small data set (easier/quicker to code & validate), but when a concern was raised about how to balance the work on the smaller dataset with the work on the larger dataset, they suggested an additional column. This demonstrated (to the observers) that the teams were leveraging the Kanban board to more than track status, but to also help strategize about how to prioritize work. Other groups adopted a simpler Kanban board, consisting of "not started", "in progress" and "done" (as opposed to the more detailed board that showed tasks across the different phases of the analysis). These groups did not show any material difference in progress, as compared to the groups that maintained the detailed Kanban board.

Why was the CRISP model effective? This was a very natural way for students to conduct the projects: understanding, analysis, etc. and making loops/iterations if necessary. For example, the teams spent their initial four weeks understanding the business requirements and the data that was available, and were the last to start coding (compared to the other process methodologies). However, since these teams delayed the analytics coding, the teams did not fully understand the coding challenges that they were going to face when they actually did start to do the analytics coding, which caused many challenges as the teams approached the project deadline.

Why was Agile Scrum not as effective? Many teams didn't create clearly defined sprints (i.e. clear / useful deliverables) and many also changed the plan during a sprint. This was due to the fact that the team was not able to estimate how long tasks would take. Task estimation was very difficult due to the exploratory nature of the work as well as the lack of experience of the students in doing data science tasks. So, the team members did not have great confidence in what could be completed within a two-week sprint, and hence, didn't define sprints in the true spirit of the methodology. This difficulty in task estimation when using scrum is similar to what has been previously reported with a data science team using scrum (Saltz, Shamshurin & Connors, 2016).

Baseline effectiveness? Perhaps as expected, the students asked for a bit of guidance from the instructor ("what should we do"), but in general were comfortable without a clearly defined methodology. This is not surprising, since from a student's perspective, this project methodology was similar to the many other projects that they had done in their other classes. As time progressed, the teams progressed in their understanding of the requirements as well as their usage of R to do the analytics. It turned out that the teams without guidance started to work in a CRISP-like methodology. In other words, they identified the phases and did several iterations. Finally, the focus on project management (e.g., asking questions about project coordination and status), without a complicated process to understand and follow, seemed to instill a focus on coordination within the team that might normally be lacking in this type of project.

6 Conlusion

In this experiment, there were clear differences in student outcomes that were driven by the different project management methodologies students were instructed to use. Given these results, instructors in data science courses, and perhaps other project-focused courses where there is uncertainty with respect to task duration estimation, should consider using an Agile Kanban methodology. However, since there were only four teams per condition, additional experiments would be helpful to confirm these findings. The increase in sample size (number of teams per condition) would enable a better understanding of the possible impact a project management methodology might have on a data science project. This would also help to alleviate possible confounding factors, such as student academic ability across the sections.

References

Ahmad, M. O., Markkula, J. and Oivo, M. 'Kanban in software development: A systematic literature review'. *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*: IEEE, 9-16.

Anderson, P., Bowring, J., McCauley, R., Pothering, G. and Starr, C. 'An undergraduate degree in data science: curriculum and a decade of implementation experience'. *Proceedings of the 45th ACM technical symposium on Computer science education*: ACM, 145-150.

Anslow, C. and Maurer, F. 'An experience report at teaching a group based agile software development project course'. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*: ACM, 500-505.

Bhardwaj, A., Bhattacherjee, S., Chavan, A., Deshpande, A., Elmore, A. J., Madden, S. and Parameswaran, A. G. 'DataHub: Collaborative Data Science & Dataset Version Management at Scale', *Conference on Innovative Data Systems Research (CIDR)*, Asilomar, California.

Borrego, M., Karlin, J., McNair, L. D. and Beddoes, K. (2013) 'Team effectiveness theory from industrial and organizational psychology applied to engineering student project teams: A research review', *Journal of Engineering Education,* 102(4), pp. 472-512.

Brunner, R. and Kim, E. (2016) 'Teaching Data Science', *Procedia Computer Science,* 80, pp. 1947-1956.

Damian, D., Lassenius, C., Paasivaara, M., Borici, A. and Schröter, A. 'Teaching a globally distributed project course using Scrum practices'. *Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD), 2012*: IEEE, 30-34.

Das, M., Cui, R., Campbell, D. R., Agrawal, G. and Ramnath, R. 'Towards methods for systematic research on big data'. *Big Data (Big Data), 2015 IEEE International Conference on*: IEEE, 2072-2081.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*, *50*(9), 833-859.

Espinosa, J. A. and Armour, F. 'The Big Data Analytics Gold Rush: A Research Framework for Coordination and Governance'. *2016 49th Hawaii International Conference on System Sciences (HICSS)*: IEEE, 1112-1121.

Gao, J., Koronios, A. and Selle, S. (2015) 'Towards A Process View on Critical Success Factors in Big Data Analytics Projects'.

Gil, Y. (2014). 'Teaching parallelism without programming: a data science curriculum for non-CS students'. *Proceedings of the Workshop on Education for High-Performance Computing*: IEEE Press, 42-48.

Grady, N. W., Underwood, M., Roy, A. and Chang, W. L. 'Big Data: Challenges, practices and technologies: NIST Big Data Public Working Group workshop at IEEE Big Data 2014'. *Big Data (Big Data), 2014 IEEE International Conference on*: IEEE, 11-15.

Jagadish, H., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R. and Shahabi, C. (2014) 'Big data and its technical challenges', *Communications of the ACM,* 57(7), pp. 86-94.

Ko, A. J., LaToza, T. D. and Burnett, M. M. (2015) 'A practical guide to controlled experiments of software engineering tools with human participants', *Empirical Software Engineering,* 20(1), pp. 110-141.

Kropp, M. and Meier, A. 'Teaching agile software development at university level: Values, management, and craftsmanship'. *2013 26th International Conference on Software Engineering Education and Training (CSEE&T)*: IEEE, 179-188.

Mariscal, G., Marban, O. and Fernandez, C. (2010) 'A survey of data mining and knowledge discovery process models and methodologies', *The Knowledge Engineering Review,* 25(02), pp. 137-166.

Ohland, M. W., Giurintano, D., Novoselich, B., Brackin, P. and Sangelkar, S. (2015) 'Supporting Capstone Teams: Lessons from Research on Motivation', *INTERNATIONAL JOURNAL OF ENGINEERING EDUCATION,* 31(6), pp. 1748-1759.

O'Neil, M. (2014) 'As data proliferate, so do data-related graduate programs', *The Chronicle of Higher Education*.

Ramamurthy, B. 'A Practical and Sustainable Model for Learning and Teaching Data Science'. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*: ACM, 169-174.

Saltz, J., Shamshurin, I., Connors, C., (2016). "A Framework for Describing Big Data Projects", in *19th International Conference on Business Information Systems.*

Saltz, J. and Heckman, R. (2016) 'Big Data science education: A case study of a project-focused introductory course', *Themes in Science and Technology Education,* 8(2), pp. 85-94.

Saltz, J. S. 'The need for new processes, methodologies and tools to support big data teams and improve big data project effectiveness'. *Big Data (Big Data), 2015 IEEE International Conference on*: IEEE, 2066-2071.

Saltz, J. S. and Shamshurin, I. 'Exploring the process of doing data science via an ethnographic study of a media advertising company'. *Big Data (Big Data), 2015 IEEE International Conference on*: IEEE, 2098-2105.

Schroeder, A., Klarl, A., Mayer, P. and Kroiß, C. 'Teaching agile software development through lab courses'. *Global Engineering Education Conference (EDUCON), 2012 IEEE*: IEEE, 1-10.

Shearer, C. (2000) 'The CRISP-DM model: The new blueprint for data mining', *Journal of Data Warehousing,* 5(4), pp. 13-22.

Vanauer, M., Bohle, C. and Hellingrath, B. 'Guiding the Introduction of Big Data in Organizations: A Methodology with Business-and Data-Driven Ideation and Enterprise Architecture Management-Based Implementation'. *System Sciences (HICSS), 2015 48th Hawaii International Conference on*: IEEE, 908-917.