Comparing Data Science Project Management Methodologies via a Controlled Experiment

Jeffrey S. Saltz
Syracuse University
jsaltz@syr.edu

Ivan Shamshurin
Syracuse University
ishamshu@syr.edu

Kevin Crowston
Syracuse University
crowston@syr.edu

Abstract

Data Science is an emerging field with a significant research focus on improving the techniques available to analyze data. However, there has been much less focus on how people should work together on a data science project. In this paper, we report on the results of an experiment comparing four different methodologies to manage and coordinate a data science project. We first introduce a model to compare different project management methodologies and then report on the results of our experiment. The results from our experiment demonstrate that there are significant differences based on the methodology used, with an Agile Kanban methodology being the most effective and surprisingly, an Agile Scrum methodology being the least effective.

1. Introduction

Data Science is an emerging discipline that combines expertise across a range of domains, including software development, data management and statistics. Data science projects typically have a goal to identify correlations and causal relationships, classify and predict events, identify patterns and anomalies, and infer probabilities, interest and sentiment [1]. Big Data is a related field, often thought of as a subset of data science, in that data science applies to large and small data sets and covers the end-to-end process of collecting, analyzing and communicating the results of the analysis.

With the increasing ability to collect, store and analyze an ever-growing diversity of data that is being generated with increasing frequency, the field of data science is growing rapidly. As a new field, much has been written about the use of data science and algorithms that can generate useful results. In fact, many in the field, such as Chen [2], believe that data science research needs to continue to focus on analytics. Unfortunately, less has been written about how a group could best work together to execute a data science project. For example, in the field of data science, there is no known "best" process to do a data science project [3].

Having a well-defined repeatable process can help data science teams across a range of challenges, including understanding who needs to be included as a stakeholder in the process, selecting an appropriate data architecture / technical infrastructure, determining the appropriate analytical techniques and validating the results. Without a well-defined process, these tasks would still likely get addressed, but the team might forget a step or not learn from their own experience and that of others, leading to a less effective process.

This paper explores the impact of different data science project methodologies within a controlled experiment, using students as subjects. The research aims to understand if one process is better than the others (with respect to what is the best methodology a team should use to do a data science project). Specifically, we focus on following two research questions:

1) How should one compare different project methodologies (is project output the only criteria)?

2) Is one project management methodology better than the other methodologies?

The rest of the paper is structured as follows. In Section 2, we discuss the prior work on data science process methodologies as well as briefly review information system experiments. Section 3 describes the methodology for our empirical study. Section 4 discusses the data we collected during and after the experiment. Section 5 presents the results from our study, and finally, section 6 summarizes the findings from our study.

2. Literature Review

While there has been some research on the challenges of doing data science, this has focused on the technical challenges in executing the projects. For example, Kaisler, Armour, Espinosa, and Money [4]

HICSS

focused on storage and data management challenges. Similarly, Katal [5] discussed the challenges caused by the rapid growth in the data that is stored and analyzed, which is outstripping the computing resources and tools available to analyze the data. Unfortunately, there has not been much written on how to ensure data teams operate effectively and efficiently.

Below, we summarize recent efforts related to methodologies of teams working on data science projects. We first discuss models describing the process of doing data science and then summarize what has been published with respect to the need for an improved methodology. Perhaps because it is a new domain, beyond what is reported below, there has been little focus on how a team could effectively work together to do data science, nor much discussion on the team-based challenges that might occur when a group of people are doing a data science project.

Finally, at the end of the literature review, we also briefly report on previous work relating to software development experiments.

2.1. The Data Science Process

With respect to data science, current descriptions of how to do data science generally adopt a task-focused approach, conveying the techniques required to analyze data. For example, Jagadish [6] described a process that includes acquisition, information extraction and cleaning, data integration, modeling, analysis, interpretation and deployment. Guo [7] approached the problem from a slightly different perspective and provided a Data Science Workflow framework. Guo's workflow defined several high-level phases such as Preparation, Analysis, Reflection, and Dissemination, with each phase having a specific series of steps that can be repeated within that phase in an iterative analysis. Interestingly, while Espinosa and Armour [8] agreed with these typical steps, they also noted that the main challenge is task coordination.

This step-by-step view on how to do data science has not materially evolved in the past 20 years. For example, they are similar to the KDD (Knowledge Discovery in Databases) process described nearly twenty years ago [9]. In another example, the *CRISP-DM* (*Cross Industry Standard Process for Data Mining)* might also be viewed as a possible first step towards defining a data science methodology. CRISP-DM was established in the 1990s, and is a data mining process model for data mining experts [10]. The model mentions six high-level phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. SEMMA [11] is the second most popular methodology, and consists of tasks such as: sampling, exploring, modifying, modeling, assessing.

While these process models differ in details, at a high level they are broadly similar. We note though that no model seems to have achieved wide acceptance. For example, there has been a reported decrease within the KDD community of people using CRISP-DM and SEMMA, and an increase in people using their own methodology [12].

Finally, it is interesting to note that the evolution on how to do data science projects might be similar to the evolution that has occurred for software development. At first, programming was thought to be a solitary task, and the work process was focused on the key steps required to create a software solution. There was an implicit assumption that the process for working across a group of people was not an issue. For example, when the classic phased software development model was defined, the process was described as a series of tasks. However, as demonstrated by the growing use of agile methodologies, it has become clear that it is useful to establish a methodology that ensures effective group communication and acknowledges that the process is iterative.

2.2. The Need for an improved Methodology

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 [3]. In a related finding, Vanauer, Bohle and Hellingrath [13] 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 [14]. Indeed, it has been argued that projects need to focus on people, process and technology [15, 16] and that task coordination is the main challenge for data projects [8].

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 [1, 15, 17]. However, the need for more guidance is recognized; e.g., a recent Gartner Consulting report advocates for more careful management of analysis processes, though a specific methodology is not identified [18].

Demonstrating the impact of this low level of process maturity, Kelly and Kaskade [19] surveyed 300 companies, and reported that "55% of Big Data projects don't get completed, and many others fall short of their objectives". While there are many reasons a project might not get completed, with a robust team-based process methodology, one would expect many of those reasons to be identified prior to

the start of the project, or to be mitigated via some aspect of the project execution and/or coordination methodology. Perhaps not surprisingly, it has been reported that an improved process model would result in higher quality outcomes [20] and at least some managers are open to improving their process methodology, but might not think of doing it unless prompted [17].

While Data science projects have parallels to other domains, there are differences as compared to these other types of projects. For example, compared to software development, data science projects have an increased focus on data, what data is needed and the availability, quality and timeliness of the data [1, 3, 21]. This suggests that the factors driving the adoption of a more mature project methodology within a data science context might be different from the factors identified in other domains. In any event, even if one argued that data science projects were similar to other information systems projects, there is clearly a current lack of adoption of mature team process methodologies for data science projects.

2.3. Software Development Experiments

While there has been little documented on data science experiments, experiments within the software development domain have been taking place for decades. According to Votta and Porter [22], empirical research in software engineering must contend with three challenges:

(1) Students vs. professionals
(2) Lab conditions vs. real life
(3) Individuals vs. groups

Students vs professionals: Students are typically used in software development experiments. For example, it was observed that students were used as subjects in 87% of the experiments analyzed over a ten year period [23]. However, it has been noted that "students vs. professionals" is actually a misrepresentation of the confounding effect of proficiency, and in fact differences in performance are much more important than differences in status [24]. Hence, using master level students, with an average of 3 years IT experience can often be a more appropriate choice than undergraduate students with minimal experience. Furthermore, when needing to compare across experimental conditions, using students may actually reduce the variability because all students have about the same level of education, leading to better statistical characteristics [25]. In addition, while students might not be as experienced as practicing software professionals, they can be viewed as the next generation of professionals and are suitable subjects for many software development experiments [26, 27].

Lab conditions vs real life: One of the challenges in creating a software development experimental design is the tradeoff between realism and control [28]. However, a task that lasts longer allows larger and more realistic tasks can improve realism while maintaining experimental control [25].

Individuals vs. groups: While there are many ways to measure an experimental outcome, the most common that are appropriate to most experiments are success on task, time on task, perceived tool/methodology usefulness and solution quality [25]. However, if one wants to compare different methodologies to improve a group's overall performance, then it becomes critical to be able to evaluate and compare the effectiveness of the different teams [3]. For example, researchers in social and organizational psychology have studied teams and their performance for decades and have many models describing and explaining team behavior and performance. Hackman's model [29] is one of the most widely used normative models and seems appropriate due to its intended purpose of identifying factors related to team effectiveness, broadly defined, and its inclusion of team process factors. In brief, this model focuses on the inputs factors (such as organizational context and group design), process and moderating factors and outputs (including task output, the team's continued capability to work together and the satisfaction of individual team members).

3. Methodology

To investigate the impact of using different project management methodologies, we conducted an experiment comparing four different process methodologies. Specifically, student teams in a master's level data science course worked on a semester long data science project, using one of four different process methodologies.

To evaluate the different project methodologies, we leveraged Hackman's model. Specifically, we held constant the input factors (such as organizational context and group design) and varied the process to be used by the different teams. Our model for team effectiveness is based on Hackman's outputs, and as shown in Figure 1, includes task output, the team's continued capability to work together and the satisfaction of individual team members.
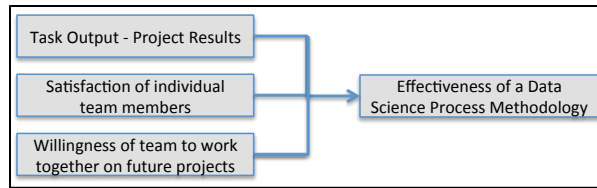
Figure 1: Process Evaluation Model

Based on this model, Table 1 shows the key measures on how we operationalized our process evaluation model. Note that in addition to the factors identified in our process evaluation model, we also observed the teams during the semester (i.e., observing the teams in action) in order to provide context to our findings as well as to try to uncover possible additional drivers of team success.

Table 1: Measuring Team Effectiveness

Key Measures	How Measured
Willingness of team to work together on future projects	End of Semester Student Survey
Satisfaction of individual team members	End of Semester Student Survey
Task output	Panel of Experts
Team dynamics and project update status	Facilitator Observations

As shown in Table 1, to measure team effectiveness and hence, address our research questions, consistent with Eisenhardt [30], we employed multiple data collection methods. In particular, our methodology was a mixed method research approach that combined qualitative and quantitative methods. Our qualitative research comprised observations and semi-structured interviews across the teams executing the projects. These interviews focused on understanding project status and team dynamics. The quantitative phase of our research involved analyzing a structured survey that was distributed to all participants at the completion of the project.

The rest of this section describes the experiment in more detail by first describing the subjects in the study, then the project that was done within the experiment. This is followed by a description of the different process methodologies (i.e., the experimental conditions) and what was held constant across the teams.

3.1. Student Subjects

A total of 85 graduate students participated in the study. Forty percent 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.

All the students that participated in the study were part of an "applied data science" course. For half the students, this was their first exposure to data science. The other half of the students had an experience level equivalent to that of an entry-level data science professional.

3.2. 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, thus the students were highly motivated to work on the project.

The project was done using 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. However, specific instructions were not provided to the students (on what analysis was to be done).

The dataset was a modified version of a real dataset of survey responses from a real hotel chain. Hence, the data was not real, but was representative of the actual challenges one might face in executing a data science project. The dataset contained roughly 3M responses to a customer survey and had a size of approximately 100GB. In all, there were 237 survey attributes (or columns) in the dataset. Students had 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, a member of their rewards program, and if so, what level), information about the hotel (ex. location) and information about the 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 (i.e., there were different survey given to different customers, some surveys asked more questions than others).

The project was positioned in a way that the students were supposed to act as consultants and analyze the large data set of customer survey responses for their client. The goal for each team was to identify and then answer "interesting" questions, such as understanding how customer satisfaction and likelihood to recommend 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. For example, what to analyze was determined by each team, and was a function of how the team determined what might be useful, what was possible with the data that was provided as well as project duration. To identify the appropriate questions, students explored the data available as well as discussed possible analysis options with their client (or more accurately, a person acting as the client).

To simulate students working within a larger organization, the students did not have access the entire dataset, but rather, made requests to the "IT department" to get a subset of the data (i.e., which columns to include, which to exclude and which to conditionally include). The results were returned in a CSV format (for easy reading into an R data frame).

3.3. Experimental Conditions

All students received in the same weekly 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 of the lab sections. Each lab section also met weekly. Within each section, students were divided into teams, with four teams, of four to six students per team. The actual 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 [25]. The weekly lab section covered practice in using hands-on data analytics as well as time for the students to work on their project.

In all, there were 16 teams, with four teams per lab section. Each lab section was taught and used a different process methodology, with respect to how to work together as a team on the data science project. 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 used to develop software systems. Specifically, the team was instructed to do "sprints" (burst of work) that last two weeks. 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 allotted for that sprint, 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 combined 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 [31] . 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, a task flowed through the defined process. Limiting the number of tasks within any one step should help to ensure the team minimizes bottlenecks and work in progress.

CRISP: Based in an industry standard, CRISP-DM [10], 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 deemed 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.

3.4. Consistency across the teams

While the different lab sections used different process methodologies to do their projects, each team was given the same project - the same data set as well as the same project goals/deliverables. In addition, each team had the same project support. The following

project support helped ensure consistency across groups working on the project.

Data Science Knowledge: All participants were instructed in data science concepts via the weekly class. In addition, students across the teams had approximately the same academic performance. While some students did have prior exposure to data science, these students were evenly distributed across the lab sections.

Process Methodology Explanation: To ensure each participant understood the methodology his or her team was to use for the project, each team was provided with a handout that explained the methodology to be used by that team. In addition to the handout, time for Q&A (with respect to the methodology) was allocated on a weekly basis. Furthermore, each team was given access to a process expert.

Access to the "client": Each team was given access to a "business champion" who was a domain expert and was the "consumer of the analysis".

Access to a Data Science expert: Each team was given access to a data science expert to guide the team from the technical perspective: what algorithms might be appropriate, what kind of pre-processing might be useful, etc.

IT / Data Support: Technical resources were available to provide data extracts from the large data repository as requested. However, this "IT Team" was a central resource, so some lead-time was required. In other words, there was a "service level agreement" (SLA), such as a maximum number of "data fetches" per week, and that the data was provided within one to three days.

4. Data Collection

To support the measures evaluating team effectiveness, that were noted in Table 1, a multi-dimensional approach was used to collect / measure team effectiveness.

First, qualitative data was obtained during the project. Specifically, each team discussed their status, findings and challenges three times during the project. For each of these updates, the student teams (of 4-6 students per team) provided a data analysis update, described next steps and were able to ask questions of their client. The goal of the three project updates was to have an honest dialog with the project team members (i.e., not try to "sugar coat" any issues in order to do well on the interim status update), thus each

project update only counted one percent of the grade. At least two faculty members observed each of the project updates, and the faculty members documented their observations for each of the student teams. Thus, there were 96 documented faculty observations (2 faculty members x 3 student team discussions x 4 teams per lab condition x 4 different lab conditions).

Second, quantitative data was collected in the analysis of the final project submission. Specifically, the project was graded on a scale of 1 to 10, with 10 being the best possible evaluation. Two faculty members did this evaluation independently. Then, to generate a score for each project, the evaluations from the two faculty members were averaged together.

Finally, 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 student like to work with their other team members 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.

5. Findings

In this section we report on the findings from the experiment. We first provide some 'during the project' observations. We next provide a comparison of the quality of the final projects (was the overall quality of the projects different, based on the experimental condition?). This is followed by an analysis of the student reported perceptions about using each of the methodologies.

5.1. Project Observations

In reviewing the observations of the student teams (that were generated during the student updates as well as by observing how the team's were actually working on the project), a couple of key themes came across. Below we describe these observations for each of the experimental conditions.

Agile Scrum: These teams started doing analytics very early, and seemed to skip most of the work that enabled the other teams to understand the client requirements and the data available. In other words, these teams typically jumped to start "doing the work" (ex. doing analytics in R), even though there was still confusion about what the client actually wanted (i.e. understanding the client requirements).

In addition, many of the teams didn't create clearly defined sprints (i.e. clear / useful deliverables) and

many also changed the plan during a sprint. This was partially due to the team members not fully understanding the methodology and partially due to the fact that the team was not able to properly estimate how long tasks would take.

Agile Kanban: The Agile Kanban teams seemed to easily use the Kanban board as a way to understand and explain their project status. In general, the teams had a good grasp of the client requirements. One team had a challenge with how granular the tasks should be on the Kanban board - they were a bit too high level. A different 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.

Some of the 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.

CRISP: 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). Their knowledge of the requirements was equal to or perhaps a bit better than the Agile Kanban teams (and much better than the Agile Scrum Teams). However, since they delayed the analytics coding, the teams did not fully understand the coding challenges they were going to face when they actually started to do the analytics until much later into the project, which caused many challenges as the teams approached the project deadline.

Baseline: Perhaps as expected, the teams 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 many others that they had done in various 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 (loop).

5.2. Project Results

Two experts independently evaluated each project (on a scale of 1 to 10, with 10 being the 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).

To compare the quality of the projects across the experimental conditions, the project scores within each condition were averaged across the expert reviewers. As shown in Table 2, teams that used the 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 statically different from the Agile Kanban and CRISP results.

Table 2: Project Results

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

5.3. Student Survey Responses

At the completion of the project, via a survey, the students were asked to agree or disagree (using a 5-level Likert scale) to several statements. For example, we explored if the team members would like to work together on future projects. Thus, as shown in Table 3 and discussed below, we investigated the key measures of team effectiveness noted previously in Table 1.

Table 3: Student Perceptions

Statement	Section	Average Score (5-level Likert)
If it was possible, I would want to work with this team on future projects	Agile Scrum	3.4
	Agile Kanban	4.2
	CRISP	4.3
	Baseline	3.8
I am very satisfied (with respect to working on this project)	Agile Scrum	3.9
	Agile Kanban	4.3
	CRISP	4.4
	Baseline	4.4
This project management method was similar to how I have done previous group projects	Agile Scrum	3.3
	Agile Kanban	2.5
	CRISP	3.3
	Baseline	3.6
It was complicated to use the project management method within my team	Agile Scrum	2.6
	Agile Kanban	3.1
	CRISP	3.0
	Baseline	2.6

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 statically different from both Agile Kanban and CRISP methodologies.

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. More experiments will help determine if this difference is statically significant.

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 statically significant in that Agile Kanban was statically 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 with respect to how complicated it was to use the methodology.

Perceptions of What Worked Well: In analyzing the more open-ended question of "what was working well", as shown in Table 4, 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.

Table 4: What Worked Well

Section	% of subjects 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 project and 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 on time successfully" "The team was coordinated"

6. Discussion

To address our first research question (how should one compare different project methodologies), we defined an initial model to evaluate different data science project management methodologies. Figure 2 shows a refined model that identifies four factors that can drive the evaluation of the different methodologies for data science projects. As expected, the quality of results of a project is an important component for evaluating projects. The willingness of the team members to work together on future projects and the satisfaction of the team members, are also key components of the model. One factor that was added to the model was how easy it was to use the methodology.
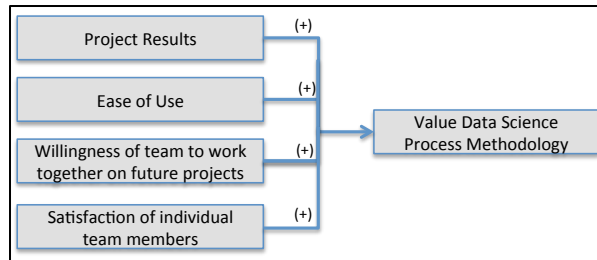
Figure 2: Enhanced Process Evaluation Model

The results of our experiment addressed our second research question (is one project management methodology better than other methodologies). In our experiment, there were two methodologies that were better than the others (Agile Kanban and the CRISP process methodologies), with Kanban being slightly better, due to Kanban's teams higher focus on teamwork (that was reported in Table 4).

Perhaps a bit surprisingly, the Agile Scrum methodology was actually worse than the Baseline condition (i.e., not giving the team members any process methodology instruction). Why not Agile Scrum? First, it was a difficult process for a team to fully utilize. In addition, task estimation was very difficult, so the team members did not have great confidence in what could be completed within a sprint. Furthermore, the methodology caused students to "jump into the data analysis" and not spend enough time understanding the data and the requirements of the clients. Finally, perhaps the methodology was difficult for students to understand and apply within a data science project context.

Why was Agile Kanban effective? Perhaps it was due to its' culture of continuous improvement. In the end, it appears Kanban helped with student teamwork skills and fostered collaboration. It also appears to have a low barrier to entry with no significant training required. We can infer this due to the fact that even though the students noted that the Kanban methodology was new to them, they seemed to embrace the Kanban methodology.

Why was the CRISP model effective? It was a very natural way of conducting the projects: understanding, analysis, etc. and making loops/iterations if necessary. It is also interesting to note that the students' without an assigned methodology started to converge into a CRISP-like methodology, even though they were not provided with any specific instructions. In other words, "baseline" teams did work in a CRISP-like manner for long parts of the project. Perhaps with this type of project, the CRISP methodology was helpful to ensure that students focused on understanding the data and the client requirements. In addition, the focus on project management, 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.

6.1. Limitations

There are several limitations with our results. First, the teams were comprised of students, which as noted earlier, have the advantage of being able to compare teams of similar backgrounds, but the disadvantage of not evaluating the methodologies with real data science teams. Hence, it is possible that industry teams, with a more varied background (i.e., senior data scientist) might require a different project methodology.

Another limitation is the actual project. This includes the fact that team members did not work on the project full-time and the fact that the data, while realistic, was not actually from a client engagement. This introduces limitations such as team member motivation and how team members prioritized this work effort as compared to other items on their plate (which is not as much of an issue within a more realistic full-time project effort).

In addition, the sample size, in terms of the number of teams evaluated, was small. More teams need to be analyzed, since one or two strong participants might have impacted the results of this experiment across the methodologies.

6.2. Next Steps

Since there were only four teams per condition, additional experiments are planned. The increase in sample size (number of teams per condition) will enable a better understanding of the possible impact a project management methodology might have on a data science project.

7. References

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

[2] H. Chen, R. H. Chiang, and V. C. Storey, "Business Intelligence and Analytics: From Big Data to Big Impact," *MIS quarterly*, vol. 36, no. 4, pp. 1165-1188, 2012.

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

[4] S. Kaisler, F. Armour, J. A. Espinosa, and W. Money, "Big data: Issues and challenges moving forward," in

System Sciences (HICSS), 2013 46th Hawaii International Conference on, 2013, pp. 995-1004: IEEE.

[5] A. Katal, M. Wazid, and R. Goudar, "Big data: issues, challenges, tools and good practices," in *Contemporary Computing (IC3), 2013 Sixth International Conference on*, 2013, pp. 404-409: IEEE.

[6] H. Jagadish *et al.*, "Big data and its technical challenges," *Communications of the ACM,* vol. 57, no. 7, pp. 86-94, 2014.

[7] P. Guo. (2013). *Data Science Workflow: Overview and Challenges*. Available: http://cacm.acm.org/blogs/blog-cacm/169199-data-science-workflow-overview-and-challenges/fulltext

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

[9] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine,* vol. 17, no. 3, p. 37, 1996.

[10] C. Shearer, "The CRISP-DM model: The new blueprint for data mining," *Journal of Data Warehousing,* vol. 5, no. 4, pp. 13-22, 2000.

[11] A. I. R. L. Azevedo and M. F. Santos, "Kdd, semma crisp-dm: a parallel overviee," *IADS-DM,* 2008.

[12] G. Piatetsky, "CRISP-DM, still the top methodology for analytics, data mining, or data science projects," October 28, 2014. Available: http://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html

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

[14] A. Bhardwaj *et al.*, "DataHub: Collaborative Data Science & Dataset Version Management at Scale," presented at the Conference on Innovative Data Systems Research (CIDR), Asilomar, California, 2014.

[15] J. Gao, A. Koronios, and S. Selle, "Towards A Process View on Critical Success Factors in Big Data Analytics Projects," 2015.

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

[17] J. S. Saltz and I. Shamshurin, "Exploring the process of doing data science via an ethnographic study of a media advertising company," in *Big Data (Big Data), 2015*

IEEE International Conference on, 2015, pp. 2098-2105: IEEE.

[18] N. Chandler and T. W. Oestreich, "Use analytic business processes to drive business performance," ed: Gartner, 2015.

[19] J. Kelly and J. Kaskade, "CIOS & BIG DATA What Your IT Team Wants You to Know ", ed, 2013

[20] G. Mariscal, O. Marban, and C. Fernandez, "A survey of data mining and knowledge discovery process models and methodologies," *The Knowledge Engineering Review,* vol. 25, no. 02, pp. 137-166, 2010.

[21] V. Dhar, "Data science and prediction," *Communications of the ACM,* vol. 56, no. 12, pp. 64-73, 2013.

[22] L. G. Votta, A. Porter, and D. Perry, "Experimental software engineering: A report on the state of the art," in *ICSE*, 1995, vol. 95, pp. 277-279.

[23] D. I. Sjøberg *et al.*, "A survey of controlled experiments in software engineering," *Software Engineering, IEEE Transactions on,* vol. 31, no. 9, pp. 733-753, 2005.

[24] Z. Soh, Z. Sharafi, B. Van den Plas, G. C. Porras, Y.-G. Guéhéneuc, and G. Antoniol, "Professional status and expertise for UML class diagram comprehension: An empirical study," in *Program Comprehension (ICPC), 2012 IEEE 20th International Conference on*, 2012, pp. 163-172: IEEE.

[25] A. J. Ko, T. D. LaToza, and M. M. Burnett, "A practical guide to controlled experiments of software engineering tools with human participants," *Empirical Software Engineering,* vol. 20, no. 1, pp. 110-141, 2015.

[26] B. A. Kitchenham *et al.*, "Preliminary guidelines for empirical research in software engineering," *Software Engineering, IEEE Transactions on,* vol. 28, no. 8, pp. 721-734, 2002.

[27] I. Salman, A. T. Misirli, and N. Juristo, "Are students representatives of professionals in software engineering experiments?," in *Proceedings of the 37th International Conference on Software Engineering-Volume 1*, 2015, pp. 666-676: IEEE Press.

[28] D. I. Sjøberg *et al.*, "Conducting realistic experiments in software engineering," in *Empirical Software Engineering, 2002. Proceedings. 2002 International Symposium n*, 2002, pp. 17-26: IEEE.

[29] J. Hackman, "The Design of Work Teams in J. Lorcsh (ed.). Handbook of Organizational Behavior: 315-342," ed: englewood cliffs, nj: prentice hall, 1987.

[30] K. M. Eisenhardt, "Building theories from case study research," *Academy of management review,* vol. 14, no. 4, pp. 532-550, 1989.

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